

# A Formal Privacy Analysis of Identity Management Systems

Meilof Veeningen      Benne de Weger      Nicola Zannone

Eindhoven University of Technology, The Netherlands  
{m.veeningen,b.m.m.d.weger,n.zannone}@tue.nl

## Abstract

With the growing amount of personal information exchanged over the Internet, privacy is becoming more and more a concern for users. In particular, personal information is increasingly being exchanged in Identity Management (IdM) systems to satisfy the increasing need for reliable on-line identification and authentication. One of the key principles in protecting privacy is data minimization. This principle states that only the minimum amount of information necessary to accomplish a certain goal should be collected. Several “privacy-enhancing” IdM systems have been proposed to guarantee data minimization. However, currently there is no satisfactory way to assess and compare the privacy they offer in a precise way: existing analyses are either too informal and high-level, or specific for one particular system. In this work, we propose a general formal method to analyse privacy in systems in which personal information is communicated and apply it to analyse existing IdM systems. We first elicit privacy requirements for IdM systems through a study of existing systems and taxonomies, and show how these requirements can be verified by expressing knowledge of personal information in a three-layer model. Then, we apply the formal method to study four IdM systems, representative of different research streams, analyse the results in a broad context, and suggest improvements. Finally, we discuss the completeness and (re)usability of the proposed method.

**Keywords:** Privacy, Identity Management, Formal methods, Data minimization, Detectability, Associability

## 1 Introduction

As communication between service providers and their customers is increasingly taking place on-line, reliable on-line identification and authentication is becoming increasingly crucial. The traditional username/password paradigm is not satisfactory: in an age of identity theft, identity fraud, phishing, and hacking, it provides too little assurance to organisations [65], while also providing little user-friendliness to customers. As novel solutions for identification and authentication emerge, this gives rise to novel privacy risks for users.

Identity management [67, 46, 35] is an emerging technology to outsource user identification, and possibly authentication, to an “identity provider”. Identity providers endorse information about their users, and provide means for authenticating a user in a service provision. To organisations, identity providers offer reduced cost for obtaining

reliable user information; to users, they offer increased convenience by letting them re-use authentication credentials.

While more business means more need for identification, it also means more privacy issues caused by the increasing amount of personal information being collected. Privacy outcries have been reported concerning both personal information being used by companies for secondary purposes [34, 73], and it being stolen and then abused by third parties [2, 40]. Both issues underline the importance of the *data minimization* principle: companies should only collect the minimal amount of personal information needed to achieve a certain purpose (and keep it only while it is needed) [57]. In several jurisdictions, privacy and data protection regulations (e.g., EU Directive 95/46/EC, HIPAA) impose stringent privacy requirements on the handling of personal information. Media attention to privacy and organisations' need to comply with regulations have spurred research interest in developing *privacy-enhancing IdM systems* [8, 26, 72], i.e., IdM systems aiming to reduce the amount of personal information that is exposed to communication partners.

Although different privacy-enhancing IdM systems have been proposed [8, 26, 72], so far there is no satisfactory way to accurately and precisely assess and compare the privacy they offer. Several works [4, 11, 41, 47] sketch privacy problems in identity management at a high level, but do not comprehensively analyse the differences between existing solutions. Other studies [1, 43] consider privacy aspects of IdM systems as part of a general comparison, but the criteria used to compare privacy are neither formal nor detailed, and thus their privacy assessments do not offer much insight into differences between systems and the reasons behind them. Existing proposals for privacy-enhancing IdM systems [8, 26, 72] assess the privacy of their own solution, but the terminology and criteria used are specific to the setting at hand, making it hard to compare different systems. In summary, currently there is no method to compare different systems in a way that is precise and verifiable, and that offers enough detail to provide real insight into the privacy differences that exist between systems.

Formal methods provide the machinery to perform such a comparison. Over the years, formal methods have arisen as an important tool to analyse security of communication in IT systems [3, 17, 52, 60]. The idea is to express communication protocols in a suitable formal model, and then verify whether such a model satisfies, e.g. authentication properties [17] or secrecy properties [9]. Secrecy, in particular, expresses one aspect of privacy; namely, whether a certain piece of information is known by some party in a protocol. However, it leaves unanswered a question which is equally important in the setting of IdM systems; namely, whether a certain piece of information can be linked to its corresponding data subject (who, in general, might not be a direct participant in the communication under analysis).

Recently, several research efforts have focused on using formal methods to analyse privacy properties [16, 31, 32, 33, 69]: in particular, privacy properties have been studied in application domains such as electronic toll collection [31], e-voting [32, 33], and RFID systems [16]. However, in some cases the properties defined and verified are specific to their respective settings [31, 32, 33]. In other cases [16], the focus is on linking messages rather than interpreting them as personal information about a data subject as needed for the analysis of IdM systems. Moreover, these methods are focused on attackers rather than authorised insiders, so they do not analyse the knowledge of different (coalitions of) actors operating within one system. In the context of IdM systems, formal methods have been used to consider privacy aspects of non-repudiation [69].

In our previous works [70, 71], we have introduced a formal method to analyse the privacy guarantees offered by protocols in which personal information is exchanged.

We introduced a three-layer model to reason about the knowledge of personal information held by different (coalitions of) communicating parties [70, 71]. The model captures that personal information in different contexts may satisfy different privacy properties; and that different pieces of information may have the same contents. We also showed how this model is determined from observations of communication between the different parties. However, while the previous works presented the basic building blocks for privacy analysis, their model only captures communication that uses a limited set of cryptographic primitives; it does not offer an implementation of the analysis method; and finally, it does not consider which privacy requirements are relevant for IdM systems.

In this work, we extend our previous analysis method to perform a formal privacy analysis of IdM systems. Specifically, our contributions are as follows:

- We present a comprehensive set of detailed privacy requirements for IdM systems, elicited through the analysis of existing systems [8, 26, 72] and taxonomies [10, 41].
- We extend our previous formal method [70, 71] for the analysis of knowledge of personal information to cover additional primitives and cryptographic protocols (specifically, zero-knowledge proofs and issuing protocols for anonymous credentials).
- We implement the analysis method in Prolog.
- We validate the proposed method by analysing four IdM systems, representative of different research streams in identity management. We show how the analysis makes it possible to compare the privacy such systems offer and to draw recommendations for their improvement.
- We discuss the main challenges concerning the (re)usability of the proposed method. Although the discussion is centred on our method, the identified challenges can be generalised to any formal method for the analysis of IdM systems.

This paper is structured as follows. First, we present an overview of identity management together with the privacy requirements to be satisfied by IdM systems, and introduce four representative systems using a case study (§2). Next, we present our formal analysis method and its Prolog implementation (§3). We then apply the formal method to the systems, and discuss the results (§4). We discuss the completeness and (re)usability of the proposed method (§5). We conclude the paper by discussing related work (§6), drawing some conclusions, and pointing to interesting directions for future work (§7).

## 2 Identity Management

In this section, we present the IdM systems we will analyse. First, we provide an overview of identity management. Then, we discuss the requirements for IdM systems relevant to privacy by data minimization. Finally, we present four representative IdM systems using a case study.

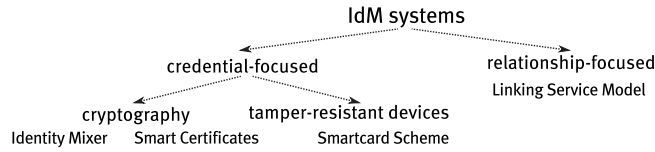


Figure 1: Taxonomy of IdM systems

## 2.1 Overview

As service providers are offering more and more customised to their users, they need to collect more and more of their personal information. Traditionally, each service provider would manage the accounts of users separately. However, this identity management model, called the *isolated user identity management model* [45], has disadvantages for both users and service providers: the user has to manually provide and update her information and keep authentication tokens for each service provider, whereas it is hard for the service provider to obtain guarantees that the information given by the user is correct.

This problem is commonly solved by delegating the task of managing and endorsing identity information to *identity providers*. Identity management is split up in two phases: *registration* and *service provision*. At registration, users establish accounts at (possibly multiple) identity providers. (This includes *identification*: i.e., the user transfers her attributes to the identity provider, and the identity provider possibly checks them. However, both the transfer and checking of attributes performed by the identity provider are out of scope of this work.) Service provision is the phase when a user requests a service from a service provider: at this point, user attributes required for the service provision need to be collected and sent to the service provider.

Identity Management (IdM) systems can be divided into two main categories [10] depending on whether or not the identity providers are involved in the service provision phase: *credential-focused* and *relationship-focused* systems (also known as network-based and claim-based systems [4]). Figure 1 shows a taxonomy of IdM systems.

In credential-focused IdM systems, the user gets long-term credentials from the identity provider in the registration phase, that she herself can present to the service providers in the service provision phase. These credentials contain her identity attributes. We can distinguish between two mechanisms employed to prevent the user from tampering with them, namely *cryptography* and *tamper-resistant devices*. Credential-focused systems relying on cryptography include CardSpace [55], U-Prove [58] and Identity Mixer [8]. The system presented in [72] relies on the use of a smart-card as a tamper-resistant device.

In relationship-focused IdM systems, in contrast, the identity provider presents the attributes to the service provider. During the registration phase, identity providers establish shared identifiers to refer to each other's identity of the user. During the service provision phase, the user authenticates to an identity provider. The identity provider then sends attributes to the service provider (possibly indirectly via the user). If needed, the shared identifiers established during registration are used to collect (or *aggregate* [26]) attributes held by other identity providers without the user having to authenticate to them as well. The combination of reliance on authentication performed by another party and exchange of identity information is sometimes referred to as *federated identity management* [45, 65]. Note that this term is also used to describe the general concept of sharing information between different domains [4] or the mere use of multiple

identity providers [1]. To avoid confusion, we will not use it further. Relationship-focused systems include Liberty Alliance [42], Shibboleth [35], and the linking service model [26].

Because in IdM systems, large amounts of personal information are processed by many different parties, privacy has become a major concern [41, 68]. In such systems, privacy threats posed by authorised insiders are nowadays considered to be a critical problem besides outsider attacks on cryptographic protocols [39]. Insiders may compile comprehensive user profiles to sell or use for secondary purposes such as marketing. These profiles can include sensitive information that is explicitly transferred by the user, but also information that is transferred *implicitly* [68]. For instance, the mere fact that a user performed a transaction at a certain service provider may be privacy-sensitive. In addition, profiles held by different parties may be combined [68] to compile even more comprehensive profiles. *Privacy-enhancing IdM systems* (e.g., [8, 26, 72]) aim to minimise the amount of information disclosed as well as prevent that different pieces of information can be linked together [41].

## 2.2 Requirements

In this section, we present and discuss the requirements for IdM systems that address privacy issues by data minimization. That is, we focus on the knowledge of personal information by insiders during normal operation of the system. The list of requirements is given in Table 1. We distinguish between requirements concerning which personal information *should* be learned (non-privacy requirements), and which personal information should *not* be learned (privacy requirements). We have elicited these requirements by analysing a number of IdM systems [8, 26, 72] as well as analysing taxonomies of privacy requirements [10, 41].

The basic requirement for IdM systems is that the service provider learns the attributes it needs [11]: *attribute exchange* (AX). Note that in one service provision, a service provider may need attributes from several identity providers.

Privacy by data minimization attempts to minimise the amount of information learned, and the extent to which it can be linked together [41]. The first aspect, information learned, can be further divided into explicitly and implicitly transferred information [68]. *Detectability* requirements capture explicitly transferred information: information about the user's attributes. *Involvement* requirements capture information about whether actors know about each other's involvement with the user: a kind of implicitly transferred personal information. The second aspect is captured in *linkability* requirements: namely, that (combinations of) parties should be able to link personal information from different sessions, databases, etc. as little as possible.

We define three detectability requirements. The first are about the service provider learning no more than strictly necessary: no attribute that he does not need to know (*irrelevant attribute undetectability*, SID), and no complete attribute value if all he needs to know is whether or not an attribute satisfies a certain property [8] (*property-attribute undetectability*, SPD). These properties limit the user profile a service provider can construct. In addition, IdM systems should guarantee that identity providers do not learn any value or property of attributes stored at other identity providers: we call this requirement *IdP attribute undetectability* (ID).

Involvement requirements address the fact that the mere interaction of a user with certain identity or service providers implies a business relation which can be privacy-sensitive. For instance, ownership of credentials can be sensitive [64] in domains such as healthcare, insurance, or finance. In addition, even if individual credentials are not

<b>Non-privacy requirements</b>	<b>Description</b>
Attribute exchange (AX)	The service provider knows the value of the required attributes/properties of the user requesting the service.
Anonymity revocation (AR)	The service provider and identity providers (possibly with help from trusted third party) can link service access to user profile.
<b>Privacy requirements</b>	<b>Description</b>
Irrelevant attribute undetectability (SID)	The service provider does not know anything about attribute values irrelevant to the transaction.
Property-attribute undetectability (SPD)	The service provider does not know anything about attributes apart from the properties he needs to know.
IdP attribute undetectability (ID)	Identity providers do not know anything about the user's attributes from other identity providers.
Mutual IdP involvement undetectability (IM)	One identity provider does not know whether a given user also has an account at another identity provider.
IdP-SP involvement undetectability (ISM)	Identity providers do not learn which service providers a user uses.
Session unlinkability (SL)	A service provider cannot link different sessions of the same user.
IdP service access unlinkability (IL)	Identity providers cannot link service access to the user profile they manage.
IdP profile unlinkability (IIL)	Collaborating identity providers cannot link user profiles.
IdP-SP unlinkability (ISL)	Identity providers and service provider cannot link service accesses to user profile at identity provider.

Table 1: Requirements for IdM systems

sensitive, the precise combination of credentials held by a user may help identify her. It is natural in identity management that the service provider learns which identity providers certify the user's attributes: this allows him to judge their correctness. However, one can aim to achieve that identity providers do not know the identity of other identity providers the user has an account at [26]: we define this as *mutual IdP involvement undetectability* (IM). In the same way, a user might want to keep hidden from her identity providers the fact that she interacts with a certain service provider: we call this requirement *IdP-SP involvement undetectability* (ISM).

Linkability is another fundamental privacy concern because it determines what user profiles can be constructed from the data that is collected [61]. To prevent a service provider from accumulating (behavioural) information, an IdM system should ensure it cannot link different service provisions to the same user: *session unlinkability* (SL). Indeed, in many cases the service provider does not need to know the identity of the user: for instance, if a user wishes to read an on-line article, the only information that is required is that she has a valid subscription.

Another concern is that parties can build more comprehensive user profiles by sharing their personal information. To prevent this, they should not know which profiles are about the same user [41]. A very strong privacy guarantee in this vein is that identity providers and service providers cannot link service provisions to the user: *IdP-SP unlinkability* (ISL). *IdP profile unlinkability* (IIL) is a weaker privacy guarantee requiring that two collaborating identity providers (without help from the service provider) can-

not link their profiles. *IdP service access unlinkability* (IL) is about the link between a service provision and the user profile at an identity provider, thus measuring whether identity providers are aware of individual service provisions.

In practice, the ISL requirement is problematic for accountability reasons: if the user misbehaves, it should be possible to identify her [8]. Several IdM systems [8, 72] introduce a trusted third party that, in such cases, can help with the identification. The *anonymity revocation* (AR) requirement states that, possibly with the help of this trusted third party, the service provider and identity providers are able to revoke the anonymity of a transaction. (Note that in particular, AR also holds if the service provider and identity providers can revoke anonymity without needing the trusted third party.)

### 2.3 Case Study

This section introduces a case study which is used to present the IdM systems studied in this paper. In particular, we consider a scenario with four main actors:

- a user: Alice, a 65 year-old woman;
- a service provider: an e-book store;
- two identity providers: one for Alice’s address (the address provider) and one for Alice’s subscription at some society (the subscription provider).

**Registration Phase** Alice creates an account at both identity providers. The address provider stores three identity attributes of the user: the street, city, and age. The subscription provider stores two user attributes: date of subscription and subscription type.

**Service provision phase** On two separate occasions, Alice purchases books from the e-book store. To this end, she needs to provide her personal information, endorsed by the identity providers, to the e-book store. The service provider, for statistical purposes, demands to know the city that Alice comes from. Moreover, the e-store offers a discount to customers that are over 60 years old. As Alice is 65 years old, she is eligible for the discount. The e-book store, however, does not necessarily need to learn her exact birth date or age; Alice can just prove that she is over 60 years old. Moreover, the e-book store does not need to know that the purchases are both made by the same user. On the other hand, in case of abuse, the service provider does want to be able to link the purchase to Alice’s profile at the address provider with the help of a trusted third party. (Note that the case study does not cover the separate issue of anonymous payment of the e-book.)

### 2.4 Four Systems

In this work, we analyse four IdM systems from the literature, representative of the types in Figure 1. We consider one traditional system, *smart certificates* [59], for whose development privacy was not a primary concern; it can be classified as credential-focused and relying on cryptography. We then consider three systems designed with privacy in mind: the *linking service model* [26], a relationship-focused IdM system; *Identity Mixer* [8], a credential-focused system relying on cryptographic protocols; and a credential-focused IdM system based on smartcards [72] we will refer to as the *Smartcard scheme*. We now briefly discuss these systems.

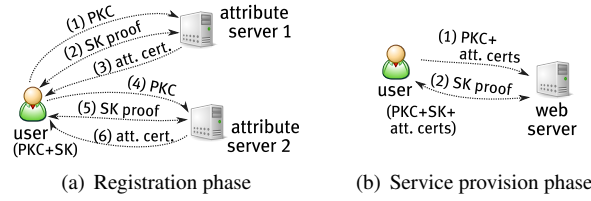


Figure 2: Smart certificates

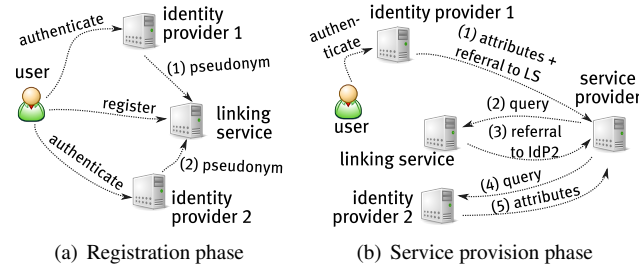


Figure 3: Linking service model

### 2.4.1 Smart Certificates

Park et al. [59] proposed an IdM system built on top of a Public Key Infrastructure (PKI). In a PKI, a certificate authority (CA) issues certificates stating that a certain public key belongs to a certain user. A user authenticates by proving knowledge of the secret key corresponding to this public key. Identity providers issue certificates that link attributes to the public key certificate. In our analysis, we consider one particular variant described in [59]: the user-pull model with long-lived certificates obtained during registration.

The flow of information is summarised in Figure 2. In the registration phase (Figure 2(a)), the user gets an attribute certificate from the identity provider (the “attribute server” in [59]), which enables her to present her attributes to others. This involves three steps: (1) the user presents her public key certificate; (2) she proves that she also knows the corresponding secret key (this is an interactive protocol shown as a two-sided arrow in the figure); and (3) the attribute server issues an attribute certificate. The process is then repeated with the other identity provider (steps (4) to (6)). The attributes in the certificate are signed using the attribute server’s secret key and hence cannot be tampered with by the user. During service provision (Figure 2(b)), the user exchanges attributes with the service provider (“web server”) in two steps: (1) she presents her public key certificate and the attribute certificates containing the attributes needed; and (2) she proves knowledge of the corresponding secret key.

The system presented in [59] is mainly designed to satisfy the attribute exchange requirement (AX) in a secure way (“the attributes of individual users are provided securely”). Privacy concerns are addressed in an extension of the system (not considered here) in which some attributes in a credential are encrypted in such a way that they can only be read by an “appropriate” server.



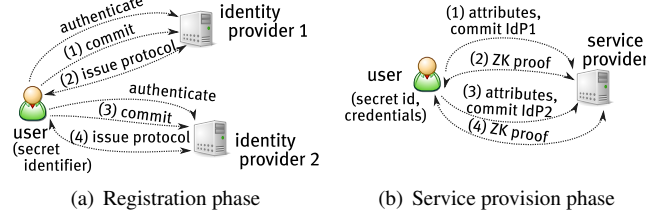


Figure 4: Identity Mixer

### 2.4.2 Linking Service Model

The linking service model [26] is a relationship-focused IdM system. Its main goal is to facilitate the collection of user attributes from different identity providers in a privacy-friendly way without the user having to authenticate to each identity provider separately. To this end, this model includes a *linking service* which is responsible for holding the links between profiles of the user at the different identity providers without knowing any personal information about the user.

The flow of information is summarised in Figure 3. During registration (Figure 3(a)), the user first creates an anonymous account at the linking service LS. LS requests the identity providers, IdP<sub>1</sub> and IdP<sub>2</sub>, to authenticate the user; each identity provider generates a pseudonym for the user and sends it to LS (steps (1) and (2)). (The specific method of authentication between the user and the identity providers and linking service is out of our scope.) In the service provision phase (Figure 3(b)), the user authenticates to IdP<sub>1</sub>. IdP<sub>1</sub> provides the service provider SP with an “authentication assertion” containing the attributes requested from it, and a referral to LS (1). The referral is an encryption of the pseudonym shared between IdP<sub>1</sub> and LS that only LS can decrypt. SP sends this referral to LS (2), which responds by sending a similar referral to IdP<sub>2</sub> (3). Finally, SP requests (4) and obtains (5) the required attributes from IdP<sub>2</sub>.

The linking service model aims to satisfy the attribute exchange requirement (AX) as well as a number of privacy requirements [26]. In particular, the main goal of the linking service model is to guarantee that identity providers do not know the involvement of other identity providers (IM). Moreover, the model aims to achieve session unlinkability (SL) through the use of random user identifiers. Finally, the linking service should not learn the partial identities of the user for the service providers; that is, it does not learn any personal information about the user. We call this requirement *LS attribute undetectability* (LD); it is not listed in Table 1 because it is only relevant for this system; however, our analysis will include the verification of this requirement.

### 2.4.3 Identity Mixer

Identity Mixer [8] is a credential-focused IdM system using a cryptographic primitive called anonymous credentials. These credentials link attributes linked to a user identifier, but are issued by identity providers and shown to service providers using protocols ensuring that neither party learns that identifier. Thus, nobody but the user knows whether different issuing or showing protocols were performed by the same user, while integrity of the attributes is still assured.

Figure 4 shows the information flows in Identity Mixer. During registration (Figure 4(a)), the user first sends a commitment to her (secret) identifier to the first identity

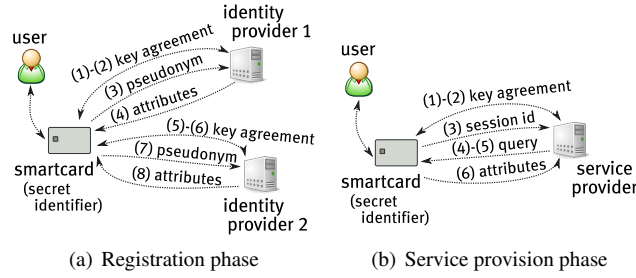


Figure 5: Smartcard scheme

provider  $\text{IdP}_1$  (1), after which the user and  $\text{IdP}_1$  together run the credential issuing protocol (2). From this, the user obtains a credential with her attributes linked to her secret identifier, without  $\text{IdP}_1$  learning the identifier. Communication with the second identity provider  $\text{IdP}_2$  is analogous (steps (3) and (4)). In the service provision phase (Figure 4(b)), the user shows information from both credentials to the service provider SP. She first shows her credential from  $\text{IdP}_1$ . To this end, she sends a message containing the attributes she wants to reveal, and “commitments” to the secret identifier and all other attributes (1). Next, she performs a zero-knowledge proof (2) which proves to SP that the attributes and commitments come from a valid credential issued by  $\text{IdP}_1$ , while revealing nothing else about the credential. The credential issued by  $\text{IdP}_2$  is shown in the same way (steps (3) and (4)).

Identity Mixer is designed to satisfy a number of privacy requirements [8]. In particular, it aims to satisfy both profile unlinkability and  $\text{IdP}/\text{SP}$  unlinkability (together called “multi-show unlinkability” in [8]) and irrelevant attribute and property-attribute undetectability (together called “selective show of data items” in [8]). The system allows for providing the service provider with an encryption of some attributes for a trusted third party (“conditional showing of data items” in [8]) that can be used for anonymity revocation. Finally, the system allows credential issuing where an identity provider copies attributes from another certificate without knowing their values (“blind certification” in [8]). The main motivation for this functionality comes from the use of these certificates for e-cash [8]. In traditional identity management scenarios, such as our case study, identity providers should know the attributes they endorse, so we do not consider this requirement in this work.

#### 2.4.4 Smartcard Scheme

Vossaert et al [72] proposed a credential-focused IdM system which relies on PKI for authentication and on smartcards (or other tamper-resistant devices) to ensure that attributes are not modified and observed during their transmission from the identity provider to the service provider. Identity providers and service providers only communicate via the smartcard, and each has a different pseudonym of the user based on a secret user identifier stored on the smartcard.

The information flow defined in the scheme is shown in Figure 5. In the registration phase (Figure 5(a)), the smartcard SC and the first identity provider  $\text{IdP}_1$  establish a secure, authenticated channel using key agreement (steps (1) and (2)). Over this secure channel, SC sends a pseudonym based on its secret identifier specific for  $\text{IdP}_1$  (3);  $\text{IdP}_1$  sends its attributes (4). Registration at the other identity provider  $\text{IdP}_2$  is similar ((steps 5) to (8)). In a service provision (Figure 5(b)), SC and service provider SP

Scheme	AX	AR	SID	SPD	ID	IM	ISM	SL	IL	IIL	ISL
Smart certificates	✓										
Linking service model	✓					✓		✓			
Identity Mixer	✓	✓	✓	✓				✓			✓
Smartcard scheme	✓	✓	✓	✓				✓		✓	✓

Table 2: Comparison of privacy requirements claimed by the various systems

establish a secure, authenticated channel as in the registration phase (steps (1) and (2)). SC generates a random session identifier (3); SP then specifies what attributes he wants, and how long they may have been cached (steps (4) and (5)). SC responds by giving the requested attributes. For anonymity revocation purposes, this response also includes Alice’s identifier encrypted for the trusted third party (6).

The system is designed to meet several requirements related to the knowledge of personal information [72]. The requirements specified correspond to our notions of attribute exchange, profile unlinkability, and anonymity revocation. Irrelevant property and property-attribute undetectability follow from their more general notion of “restraining released personal data”. The Smartcard scheme also aims to fulfil IdP profile unlinkability and IdP/SP unlinkability by preventing collusion of identity and service providers.

#### 2.4.5 Privacy Requirements Claimed by Systems

Table 2 summarises the privacy claims for the systems. In this work, we will formally verify whether these claims actually hold. In addition, we will analyse the systems against the complete range of identified requirements in order to achieve a comprehensive comparison of their privacy features. The formal methods presented in the next section will allow us both to verify claimed requirements, and to find out which non-claimed requirements still hold.

### 3 Formal Analysis of Personal Information Knowledge

This section presents a formal framework for the analysis of IdM systems, extending the work in [70, 71]. In [70], we described how to express knowledge of personal information in terms of items and links between them. In [71], by extending this idea, we modelled the knowledge of personal information arising from observed messages using a limited set of cryptographic primitives.

In this paper, we extend this model by considering properties of user attributes, additional primitives (digital signatures with appendix, labelled encryption, authenticated key exchange, and anonymous credentials), and cryptographic protocols (zero-knowledge proofs and issuing protocols for anonymous credentials). In addition, by introducing traces, we formalise knowledge evolution by the transmission of messages between different parties. Finally, we present a Prolog implementation of the above formal model.

#### 3.1 Three-Layer Model

We now present a formal model of actor knowledge.

### 3.1.1 Personal Information

A piece of personal information in the digital world is a *specific* string that has a *specific* meaning as personal information about a *specific* person. We distinguish between two types of digital personal information: identifiers and data items. Identifiers are unique within the system; for data items this is not necessarily the case. The sets of identifiers and data items are denoted  $\mathcal{I}$  and  $\mathcal{D}$ , respectively. The set  $\mathcal{E}$  of *entities* models the real-world persons whom the considered information is about.

The link between the information and its subject is captured by the *related* relation, denoted  $\leftrightarrow$ . This is an equivalence relation on entities, identifiers and data items, such that  $o_1 \leftrightarrow o_2$  means that  $o_1$  and  $o_2$  are information about the same person. In particular, any identifier or data item is related to exactly one entity. Elements of the set  $\mathcal{O} := \mathcal{E} \cup \mathcal{I} \cup \mathcal{D}$  are called *items of interest*.

We give additional structure to the above sets in two respects. First, private and public keys are particular cases of identifiers: they form sets  $\mathcal{K}^- \subset \mathcal{I}$  and  $\mathcal{K}^+ \subset \mathcal{I}$ , respectively. Given a private key  $k^-$ , the corresponding public key is  $k^+$  and vice versa. Second, we express that data items satisfy certain “properties” from a fixed set  $\{\psi_1, \dots, \psi_n\}$  relevant to the application domain. Suppose that  $\psi_i$  represents the property stating that an age is over 60. Then,  $\psi_i(\text{age}_c) \in \mathcal{D}$  expresses that  $\text{age}_c$  is an age that is over 60;  $\psi_i(\text{age}_d) \notin \mathcal{D}$  and  $\psi_i(\text{city}_c) \notin \mathcal{D}$  states that the property does not hold (or it has no meaning) for the given data item.

These concepts, however, are insufficient to model all requirements for IdM systems we are interested in. Several requirements in Table 1 are about whether or not an actor (or set of actors) can “link” two pieces of personal information to each other, i.e., whether or not the actor knows they are related. However, expressing this becomes problematic when an actor learns the same piece of information in two different contexts without realizing that it is the same information. For instance, in our case study, the e-book store will learn two profiles containing data items  $\text{city}_{al}$ ,  $\psi_i(\text{age}_{al})$ . To express session unlinkability, the model should be able to express the difference between the “instances” learned in the first and second profile: the store can link the instance of  $\text{city}_{al}$  in profile 1 to the instance of  $\psi_i(\text{age}_{al})$  in profile 1, but not to the instance in profile 2. Thus, the above model needs to be extended to distinguish between different instances of the same piece of personal information.

In addition, an actor may be able to deduce information from the fact that different pieces of information have the same string contents. For instance, suppose that in a service provision, the user’s identifier is transmitted using deterministic encryption. Then, although an observer may not be able to determine the identifier, he still knows that different service provisions involved the same user.

### 3.1.2 Three-Layer Model

Because of the need to distinguish different instances of the same piece of information, but also to reason about message contents, we introduce a three-layer representation of personal information. The representation consists of the *object layer*, *information layer*, and *contents layer*. At the information layer, as described above, the information itself is represented, e.g., “Alice’s city”. At the object layer, information is described in terms of the context in which it has been observed, e.g., “the city of the user in service provision #1”. At the contents layer, information is described in terms of the strings actually transmitted in a protocol, e.g., “Eindhoven”.

At the object layer, we model the *context* in which an actor knows pieces of in-

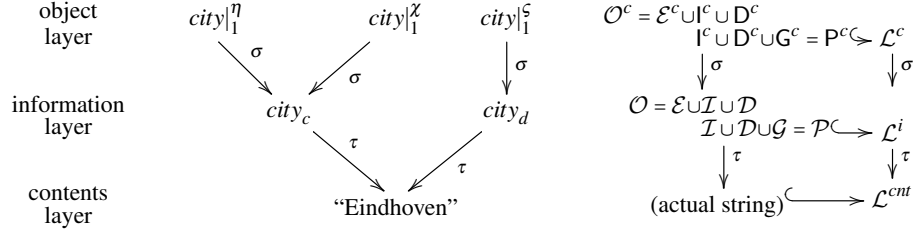


Figure 6: Example of the three-layer model: three different context items with the information and contents they represent (left); the three-layer model of information (right)

formation. A context is a tuple  $(\eta, k)$ , where  $\eta$  is a *domain* and  $k$  is a *profile* within that domain. A domain is any separate digital “place” where personal information is stored or transmitted; in our case study, the databases held by the identity and service providers are domains; also every instance of a communication protocol can be seen as a domain. Profiles contain personal information about one user in a domain. For instance, databases contain several user profiles; moreover, every logical role in a protocol instance is a different profile (though different roles could be performed by the same entity, e.g. an identity provider that also acts as service provider).

In such a context, pieces of information are represented by *variables*. This representation makes it possible to reason about such personal information without regarding the instantiation. *Identifier variables* represent identifiers (set  $\mathcal{I}$ ), whereas *data item variables* represent data items (set  $\mathcal{D}$ ): e.g., a variable  $city \in \mathcal{D}$  may denote the city in a profile. A *context data item* is a data item variable  $d$  in a context  $(\eta, k)$ , denoted  $d|_k^\eta \in \mathcal{D}^c$ ; the set  $\mathcal{I}^c$  of *context identifiers* is defined similarly. Entities are not represented by variables; instead, an entity  $e \in \mathcal{E}$  in a context  $(\eta, k)$  is denoted  $e|_k^\eta$ ; the set of *context entities* is  $\mathcal{E}^c$ . The reason is that, because entities are not digital information, there cannot be multiple “instances” of an entity. Every context contains exactly one entity who is the data subject, i.e., all information in the context is about that entity.  $\mathcal{O}^c := \mathcal{E}^c \cup \mathcal{I}^c \cup \mathcal{D}^c$  is the set of *context items of interest*. The structure of private/public keys and properties transfers to the object layer: context private and public keys are modelled by sets  $\mathcal{K}_c^-, \mathcal{K}_c^+ \subset \mathcal{I}^c$ ; a property  $\psi_i$  holding for a context data item  $d$  is denoted  $\psi_i(d)$ .

Items at the contents layer can be seen as strings of arbitrary length in some alphabet, i.e., the set  $\Sigma^*$ . The exact form of the contents layer is not relevant for our purposes. Rather, it is relevant to determine whether two pieces of information have the same contents: this is expressed using the  $\tau$  function, as described below.

### 3.1.3 Maps Between Layers and Equivalence

The link between the object layer and the information layer is given by the *substitution* function  $\sigma : \mathcal{O}^c \rightarrow \mathcal{O}$ . This function satisfies the following properties:

1.  $\sigma(\mathcal{D}^c) \subset \mathcal{D}$ ,  $\sigma(\mathcal{I}^c) \subset \mathcal{I}$ ; for any entity  $e$ , context  $(\eta, k)$ :  $\sigma(e|_k^\eta) = e$ ;
2.  $\sigma(x|_k^\eta) \leftrightarrow \sigma(y|_k^\eta)$  for any context items  $x|_k^\eta, y|_k^\eta$ ;

3.  $\sigma(K_c^+) \subset \mathcal{K}^+$ ,  $\sigma(K_c^-) \subset \mathcal{K}^-$ , and  $\sigma(key^+|_k^\eta) = k^+$  if and only if  $\sigma(key^-|_k^\eta) = k^-$ ;
4.  $\psi_i(d) \in D^c$  iff  $\psi_i(\sigma(d)) \in \mathcal{D}$ .

Intuitively,  $\sigma$  maps: 1. context items to information items of the corresponding type; 2. context items from the same context to related items of interest; 3. private/public key context items to private/public keys; 4. property context items to properties.

The link between information and its contents is given by the function  $\tau$ . The domain of the function is  $\mathcal{I} \cup \mathcal{D}$  (entities have no contents). The function  $\tau$  satisfies two properties. First, it is injective on  $\mathcal{I}$ : this formally expresses the uniqueness of identifiers within the system. Second, if  $\psi_i(d), \psi_i(d') \in D^c$ , then  $\tau(\psi_i(d)) = \tau(\psi_i(d'))$ ; that is, the contents of an attribute property are independent from the attribute value.

We introduce notation for context items  $x|_k^\eta, y|_l^\chi$  representing the same information or contents. If  $\sigma(x|_k^\eta) = \sigma(y|_l^\chi)$ , then we write  $x|_k^\eta \equiv y|_l^\chi$  and we call  $x|_k^\eta$  and  $y|_l^\chi$  *equivalent*. If  $\tau(\sigma(x|_k^\eta)) = \tau(\sigma(y|_l^\chi))$ , then we write  $x|_k^\eta \doteq y|_l^\chi$  and we call them *content equivalent*. Clearly, equivalence implies content equivalence. Because  $\tau$  is injective on identifiers, two identifiers are equivalent iff they are content equivalent.

**Example 1.** Consider three context data items  $city|_1^\eta, city|_1^\chi$ , and  $city|_1^\zeta$  (Figure 6, left side) where  $city \in D$ . Let  $\sigma(city|_1^\eta) = \sigma(city|_1^\chi) = city_c$ ,  $\sigma(city|_1^\zeta) = city_d$ , and  $\tau(city_c) = \tau(city_d) = \text{“Eindhoven”}$ . Then,  $city|_1^\eta$  and  $city|_1^\chi$  are equivalent; moreover, all three context messages are content equivalent.  $\square$

### 3.2 Actor Knowledge: Detectability and Associability

We now show how knowledge of personal information by an actor, or by a coalition of actors, is derived from messages he has observed. An actor is an entity with a view on the system; the set of actors in the system is denoted  $\mathcal{A} \subset \mathcal{E}$ . The requirements in Table 1 can be analysed by determining (1) what context items the actor(s) can *detect*; and (2) which of these items the actor(s) can *associate*. A basic version of the following analysis method is described in [70]; we improve it by modelling additional cryptographic primitives: labelled encryption, zero-knowledge proofs, authenticated key agreement, and anonymous credentials and their issuing protocols. We also model reasoning about attribute properties. Finally, we model digital signatures with appendix [53], whereas [70] models clear-signing. These primitives are necessary to analyse the selected IdM systems.

#### 3.2.1 Communication Messages

Communication in identity management protocols involves messages built up from personal and other information using cryptographic primitives such as encryption, signatures, and hashes. We model these messages by means of the set  $\mathcal{L}^c$  of *context messages*. The basic building blocks of messages are context data items  $D^c$  and identifiers  $I^c$ , and non-personal information, such as shared keys and nonces, from  $G^c$ . Items in  $G^c$  belong to a domain, but not to a user profile: in this case we denote the profile as  $\cdot$ , e.g.  $shakey|^\eta$ . The three sets together form the set  $P^c := D^c \cup I^c \cup G^c$  of *context items*.

The set  $\mathcal{L}^c$  is built up from  $P^c$  with the grammar shown in Table 3. The concatenation, hash, and (a)symmetric encryption primitives are standard [29, 37]. Digital signatures are “with appendix” [53]: that is, an actor needs to know the message that was signed in order to verify the signature. Labelled asymmetric encryption [8] is asymmetric encryption to which a label is unmodifiably attached at encryption time.

Messages	Meaning
$M, M_i ::= \emptyset \mid p \mid$	empty message, atomic message
$\{M_1, M_2\} \mid$	(associative) concatenation of messages $M_1$ and $M_2$
$S_{k^-}(M) \mid$	digital signature of message $M$ with private key $k^-$
$\mathcal{H}(M) \mid$	hash of message $M$
$E'_{M_1}(M_2) \mid$	symmetric encryption of message $M_2$ with key $M_1$
$E_{k^+}(M) \mid$	asymmetric encryption of message $M$ with public key $k^+$
$E_{k^+}(M_1)_{M_2} \mid$	labelled asymmetric encryption of message $M_1$ with public key $k^+$ and label $M_2$
$\text{AKA}(k_1^-; M_1; k_2^-; M_2) \mid$	derived key from authenticated key agreement (AKA) with (SK, randomness) pairs $(k_1^-, M_1)$ and $(k_2^-, M_2)$
$\text{cred}_{k^-}^{M_1}(M_2; M_3) \mid$	anonymous credential with user identifier $M_1$ , issuer private key $k^-$ , attributes $M_2$ , and randomness $M_3$
$\text{ZK}(M_1; M_2; M_3; M_4) \mid$	zero-knowledge proof of knowledge of secret $M_1$ with properties $M_3$ using public information $M_2$ and randomness $M_4$
$\text{ICred}_{k^-}^{M_1}(M_2; M_3)$	issuing protocol for anonymous credential $\text{cred}_{k^-}^{M_1}(M_2; M'_3)$ , where $M'_3$ is derived from $M_3$

Table 3: The grammar  $\mathcal{L}^c$  of context messages:  $p \in P^c$ ;  $k^+ \in K_c^+$ ;  $k^-, k_i^- \in K_c^-$ ;  $\emptyset$  is an empty message.

For instance, the label can represent a policy specifying when the recipient is allowed to decrypt the data.

Authenticated key agreement (AKA) [49] allows two parties to derive a unique session key based on secret keys and randomness contributed by both parties. We consider the variant presented in [49] in which both parties send each other a random value. Both parties can determine the session key, modelled by the AKA primitive, from one private key, the other public key, and the randomness.

Finally, the *cred* primitive models anonymous credentials [8]. An anonymous credential  $\text{cred}_{k^-}^{M_1}(M_2; M_3)$  represents an endorsement with private key  $k^-$  that the attributes  $M_2$  belong to the user with identifier  $M_1$ , randomised using  $M_3$ . Using the protocols we describe next, anonymous credentials can be issued without the issuer obtaining the credential or learning  $M_1$ ; also, their ownership can be proven efficiently without revealing the credential itself.

Several two-party cryptographic protocols occur in the systems presented in Section 2.4. These protocols only have meaning when looked at as a whole, i.e., the meaning lies not in individual messages, but in their combination in a particular order. Thus, we model the complete transcript (i.e., all messages of all participants) of such a protocol as one primitive. We introduce two such primitives.

First, we model a family of zero-knowledge (ZK) proofs (e.g., [30]) by means of the ZK primitive. In a ZK proof for a given property, a prover wants to convince a verifier that he knows some secrets satisfying that property with respect to some given public information, without revealing anything about the secrets. Here, we consider ZK proofs proving that (1) the public information has a certain message structure with respect to the private information, and (2) some secret attributes  $d_i$  have some properties  $\psi_k(d_i)$  to be verified. For instance,  $\text{ZK}(\{d, n\}; \mathcal{H}(\{d, n\}); \psi_2(d); n')$  denotes a ZK proof (using randomness  $n'$ ) convincing a verifier knowing the hash  $\mathcal{H}(\{d, n\})$  that the prover knows the pre-image  $\{d, n\}$  of the hash, and that  $\psi_2(d)$  is satisfied; without the verifier learning anything else about  $d$  or  $n$ . See Appendix A.1 for a detailed discussion.

Second, we model the issuing protocol for anonymous credentials [8] by means of

<b>Axiom</b> ( $\vdash 0$ )	$\frac{}{\mathcal{C}_a \vdash m} (m \in \mathcal{C}_a) \quad (\vdash 0)$	<b>Prop</b> ( $\vdash * \psi$ )	$\frac{\mathcal{C}_a \vdash d}{\mathcal{C}_a \vdash \psi_i(d)} (\psi_i(\sigma(d)) \in \mathcal{D}) \quad (\vdash E\psi)$
	<b>Testing</b> ( $\vdash T$ )	$\frac{\mathcal{C}_a \vdash m \quad \mathcal{C}_a \vdash n' \quad (m \Rightarrow n; \quad n \doteq n')}{\mathcal{C}_a \vdash n} \quad (\vdash T)$	
<b>Content analysis</b> ( $\vdash C$ )	$\frac{\mathcal{C}_a \vdash n_1 \quad \mathcal{C}_a \vdash m_1 \quad \mathcal{C}_a \vdash m_2 \quad ((m_1 \doteq m_2) \Rightarrow (m_3 \doteq m_4)); \quad n_1 =_{m_3 \sim m_4} n_2}{\mathcal{C}_a \vdash n_2} \quad (\vdash C)$		

Figure 7: Deductive system: general rules ( $\mathcal{C}_a$  a set of context messages,  $m, m_i, n, n'$ ,  $n_i$  context messages;  $d$  a context data item;  $n_1 =_{m_3 \sim m_4} n_2$  means  $n_1$  and  $n_2$  are equal up to replacing  $m_3$  by  $m_4$  and vice versa)

the *ICred* primitive. This protocol is run between a user and an issuer. In advance, both parties need to know the attributes to be certified, but only the user needs to know the identifier to which the attributes are issued. As a result of the protocol, the user obtains an anonymous credential linking the attributes to the identifier. The issuer does not learn the credential; moreover, because he does not know the identifier, he cannot issue credentials in her name without her involvement. Also, by using ZK proofs for proving ownership, the identifier is kept secret with respect to the service provider, thus preventing linkage. See Appendix A.2 for details.

Analogously to messages at the object layer, we also consider messages at the information and contents layers, and extend  $\sigma$  and  $\tau$  to messages. Thus,  $\sigma$  maps grammar elements of  $\mathcal{L}^c$  to elements of  $\mathcal{L}^i$  defined by the same grammar, but instead generated by  $\mathcal{P} = \mathcal{D} \cup \mathcal{I} \cup \mathcal{G}$ , where  $\mathcal{G}$  is the information represented by items in  $\mathcal{G}^c$ . E.g., if  $\sigma(city|_1^\eta) = city_c$  and  $\sigma(city|_1^\zeta) = city_d$ , then  $\sigma(\{\mathcal{H}(city|_1^\eta), city|_1^\zeta\}) = \{\mathcal{H}(city_c), city_d\}$ . A similar set  $\mathcal{L}^{cnt}$  and map  $\tau: \mathcal{L}^i \rightarrow \mathcal{L}^{cnt}$  are defined for contents. When contexts, domains, and profiles are applied to messages, they apply to all context items in the message, e.g.,  $E_{shake|_1}((city|_1)^\eta) := E_{shake|_1}^\eta(city|_1^\eta)$  and  $\{id, city\}_1^\eta := \{id|_1^\eta, city|_1^\eta\}$ . Like context items, context messages  $m$  and  $n$  are *equivalent* iff  $\sigma(m) = \sigma(n)$ , and *content equivalent* iff  $\tau(\sigma(m)) = \tau(\sigma(n))$ . The three-layer model of messages is shown in Figure 6 (right side).

Messages in our model satisfy two important properties: their contents are *deterministic* and *unique*. By deterministic, we mean that given the same contents as input, cryptographic primitives always give the same output (i.e.,  $\tau$  is well-defined). Randomness, e.g. in signing or in non-deterministic encryption, can be modelled explicitly as part of the plaintext. This makes it possible to distinguish the case where an actor observes two different randomised encryptions with the same input from the case where he observes the same randomised encryption twice; in the latter case, we will allow an actor to draw certain conclusions from this. Uniqueness is expressed by the *structural equivalence* assumption [71]. Note that elements of  $\mathcal{L}^{cnt}$  could a priori be the same as strings, e.g. a collision in the hash function could cause  $\mathcal{H}(\tau(x))$  and  $\mathcal{H}(\tau(y))$  to be the same string even if  $\tau(x) \neq \tau(y)$ ; or  $E'_{\tau(x)}(\tau(y))$  could happen to be the same string as  $\mathcal{H}(\tau(z))$ . We assume that this does not happen, i.e., the grammar  $\mathcal{L}^{cnt}$  uniquely represents message contents. (Of course, different elements of  $\mathcal{L}^i$  can still map to the same element in  $\mathcal{L}^{cnt}$  if the information items have the same contents.)



$\frac{\text{Hash } (\vdash^* \mathbf{H}) \quad \frac{C_a \vdash m}{C_a \vdash \mathcal{H}(m)} (\vdash \mathbf{CH})}{C_a \vdash E'_n(m) \quad C_a \vdash n} (\vdash \mathbf{EE})$		$\frac{\text{Symmetric encryption } (\vdash^* \mathbf{E}) \quad \frac{C_a \vdash m \quad C_a \vdash n}{C_a \vdash E'_n(m)} (\vdash \mathbf{CE})}{C_a \vdash m \quad C_a \vdash n} (\vdash \mathbf{CC})$	
$\frac{C_a \vdash \{m, n\}}{C_a \vdash n} (\vdash \mathbf{EC}')$		$\frac{\text{Concatenation } (\vdash^* \mathbf{C}) \quad \frac{C_a \vdash m \quad C_a \vdash n}{C_a \vdash \{m, n\}} (\vdash \mathbf{CC}) \quad \frac{C_a \vdash \{m, n\}}{C_a \vdash m} (\vdash \mathbf{EC})}{C_a \vdash E_{k^+}(m) \quad C_a \vdash k^-} (\vdash \mathbf{EA})$	
$\frac{C_a \vdash E_{k^+}(m) \quad C_a \vdash k^-}{C_a \vdash m} (\vdash \mathbf{EA})$		$\frac{\text{Asymmetric encryption } (\vdash^* \mathbf{A}) \quad \frac{C_a \vdash m \quad C_a \vdash k^+}{C_a \vdash E_{k^+}(m)} (\vdash \mathbf{CA})}{C_a \vdash m \quad C_a \vdash k^+ \quad C_a \vdash n} (\vdash \mathbf{CL})$	
$\frac{C_a \vdash E_{k^+}(m)_n}{C_a \vdash n} (\vdash \mathbf{EL})$		$\frac{\text{Labelled asymm. encryption } (\vdash^* \mathbf{L}) \quad \frac{C_a \vdash m \quad C_a \vdash k^+ \quad C_a \vdash n}{C_a \vdash E_{k^+}(m)_n} (\vdash \mathbf{CL})}{C_a \vdash E_{k^+}(m)_n \quad C_a \vdash k^-} (\vdash \mathbf{EL}')$	
$\frac{C_a \vdash E_{k^+}(m)_n \quad C_a \vdash k^-}{C_a \vdash m} (\vdash \mathbf{EL}')$		$\frac{\text{Sign } (\vdash^* \mathbf{S}) \quad \frac{C_a \vdash m \quad C_a \vdash k^-}{C_a \vdash S_{k^-}(m)} (\vdash \mathbf{CS})}{C_a \vdash \{k_1^+, n_1, k_2^-, n_2\}} (\vdash \mathbf{CG})$	
$\frac{\text{Auth Key Agr } (\vdash^* \mathbf{G}) \quad \frac{C_a \vdash \{k_1^-, n_1, k_2^+, n_2\}}{C_a \vdash \text{AKA}(k_1^-; n_1; k_2^-; n_2)} (\vdash \mathbf{CG})}{C_a \vdash \text{AKA}(k_1^-; n_1; k_2^-; n_2)} (\vdash \mathbf{CG}')$		$\frac{C_a \vdash \{k_1^+, n_1, k_2^-, n_2\}}{C_a \vdash \text{AKA}(k_1^-; n_1; k_2^-; n_2)} (\vdash \mathbf{CG}')$	

Figure 8: Deductive system: inference rules for basic primitives ( $C_a$  a set of context messages;  $m, n, n_*$  context messages;  $k^+/k^-$  and  $k_*^+/k_*^-$  public/private key pairs)

### 3.2.2 A Deductive System For Detectability

We now formalise what context items an actor (or a coalition of actors) can detect. For actor  $a \in \mathcal{A}$  knowing context messages  $C_a \subset \mathcal{L}^c$ ,  $C_a \vdash m$  means that  $a$  can *derive* context message  $m$ . The semantics of  $\vdash$  is defined by means of the deductive system given in Figures 7, 8, 9, and 10. Context items  $p \in \mathcal{P}^c$  such that  $C_a \vdash p$  are called *detectable* by  $a$ . For a coalition  $A \subset \mathcal{A} = \{a_1, \dots, a_n\}$  of actors,  $\vdash$  is applied to the union  $C_A = C_{a_1} \cup \dots \cup C_{a_n}$  of their respective message sets.

Deductive systems are commonly used in protocol analysis to reason about what messages an attacker can fabricate (e.g., [29, 37]). For this, the context in which a message is known is not relevant, and thus not considered: i.e., existing deductive systems operate at our information layer. Conversely, for our purposes the context *is* relevant; hence we extend standard deductive systems to the object layer.

Figure 7 presents the rules of the deductive system that are not about particular primitives.  $(\vdash \mathbf{0})$  is the standard axiom to derive known messages. The  $(\vdash \mathbf{E}\psi)$  rule allows the derivation of properties of attributes (if the attribute indeed satisfies the property). The testing rule  $(\vdash \mathbf{T})$  and content analysis rule  $(\vdash \mathbf{C})$  are needed to mimic actors' reasoning about different representations of information. For instance, suppose an actor knows the key for decrypting a certain message, but not in the message's context. By trying out the key, he can generally find out it is a valid decryption key, learning the key in the context of the message. The testing rule captures this situation. The content analysis rule captures several other kinds of conclusions drawn from observing the same message contents in different contexts. They are discussed in detail at the end of this subsection.

Figures 8, 9, and 10 model the primitives from Table 3 by *construction* and *elimination* rules as in traditional deductive systems [29, 37]. Construction rules express construction of a message from its components. For instance, an encryption can be constructed from the plaintext and key. Elimination rules express recovery of the components of a message. For instance, an actor can recover the plaintext from an encryption if he knows the key.

$$\begin{array}{c}
\frac{\mathcal{C}_a \vdash \{m_1, \dots, m_j, n_1, \dots, n_k, p_1, \dots, p_l, n_p, n_v\}}{\mathcal{C}_a \vdash \text{ZK}(\{m_1, \dots, m_j\}; \{n_1, \dots, n_k\}; \{p_1, \dots, p_l\}; \{n_p, n_v\})} (\vdash \text{CZ}) \\
\\
\frac{\mathcal{C}_a \vdash \text{ZK}(m_1; m_2; m_3; \{n_p, n_v\})}{\mathcal{C}_a \vdash m_3} (\vdash \text{EZ}_1) \\
\\
\frac{\mathcal{C}_a \vdash \{\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}), m_2, n_p\}}{\mathcal{C}_a \vdash m_1} (\vdash \text{EZ}_2)
\end{array}$$

Figure 9: Deductive system: inference rules for ZK proofs ( $\mathcal{C}_a$  a set of context messages;  $m_*$ ,  $n_*$  context messages;  $p_i$  properties of  $m_k$ , i.e., every  $p_i = \psi_j(m_k) \in D^c$  for some  $j, k$ )

$$\begin{array}{c}
\frac{\mathcal{C}_a \vdash \{k^-, m_1, m_2, n\}}{\mathcal{C}_a \vdash \text{cred}_{k^-}^{m_1}(m_2; n)} (\vdash \text{CR}) \quad \frac{\mathcal{C}_a \vdash \{k^-, m_1, m_2, n\}}{\mathcal{C}_a \vdash \text{ICred}_{k^-}^{m_1}(m_2; n)} (\vdash \text{CI}) \\
\\
\frac{\mathcal{C}_a \vdash \{\text{ICred}_{k^-}^{m_1}(m_2; \{n_i\}_{i=1}^7), \mathcal{H}(m_1, n_1), k^+, n_3\}}{\mathcal{C}_a \vdash \{m_1, n_1, n_2\}} (\vdash \text{EI}_1) \\
\\
\frac{\mathcal{C}_a \vdash \{\text{ICred}_{k^-}^{m_1}(m_2; \{n_i\}_{i=1}^7), k^+, m_2, n_6\}}{\mathcal{C}_a \vdash k^-} (\vdash \text{EI}_2) \quad \frac{\mathcal{C}_a \vdash \{\text{ICred}_{k^-}^{m_1}(m_2; \{n_i\}_{i=1}^7), n_2\}}{\mathcal{C}_a \vdash \text{cred}_{k^-}^{m_1}(m_2; \{n_2, n_5\})} (\vdash \text{EI}_3)
\end{array}$$

Figure 10: Deductive system: inference rules for anonymous credentials ( $\mathcal{C}_a$  a set of context messages;  $m_i$ ,  $n$ ,  $n_i$  context messages;  $k^-$  context private key)

Figure 8 covers several basic primitives. Hashes, symmetric encryption, concatenation, signature, and (labelled) asymmetric encryption are modelled as usual [29]. For labelled encryption, note that the label can be derived from the encryption ( $\vdash \text{EL}'$ ), but to change it, the plaintext is needed, i.e., the label is unmodifiably attached. To derive a session key using authenticated key agreement, an actor needs to know one of the private keys used, the other public key, and both parties' randomness ( $\vdash \text{CG}$ ), ( $\vdash \text{CG}'$ ).

Cryptographic protocols are also modelled by construction and elimination rules. The primitive represents the complete protocol transcript. The construction rule represents one actor simulating the whole protocol run; the usual case when two actors run the protocol together and both contribute inputs is captured by traces (Section 3.3). Although both actors involved learn the same protocol transcript, what information they can derive from it will in general depend on the other knowledge they have.

Figure 9 models privacy aspects of a large family of ZK proofs known as “ $\Sigma$ -protocols” [30]. There are  $\Sigma$ -protocols for many properties; in particular, they are used to prove properties of anonymous credentials [8]. The randomness for  $\Sigma$ -protocols is of the form  $\{n_p, n_v\}$ , representing contributions by the prover and verifier, respectively. Apart from the usual construction rule, there are two elimination rules: ( $\vdash \text{EZ}_1$ ) states that the property proven by a ZK proof can be seen from its transcript; ( $\vdash \text{EZ}_2$ ) states that the prover's secret may be derived from the public information and the prover's randomness. We assume that parties do not re-use their randomness; also, because we are only interested in privacy aspects, we have not included rules to derive randomness used in the ZK proof. See Appendix A.1 for details.

Figure 10 models anonymous credentials and their issuing protocol based on SRSA-

Testability ( $* \Rightarrow *$ )	Testing by...
$E'_n(m) \Rightarrow n$	Trying decryption with key
$E_{k^+}(m) \Rightarrow k^-$	Trying decryption with key
$E_{k^+}(m)_n \Rightarrow k^-$	Trying decryption with key
$S_{k^-}(m) \Rightarrow \{k^+, m\}$	Signature verification
$\text{cred}_{k^-}^{m_1}(m_2; n) \Rightarrow \{k^+, m_1, m_2\}$	Signature verification (§ A.2)
$\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \Rightarrow m_2$	Verification of ZK proof
$\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \Rightarrow n_p$	Re-calculating commitment
$\text{ICred}_{k^-}^{m_1}(m_2; \{n_i\}_{i=1}^7) \Rightarrow \dots$	
... $\{m_1, k^+, n_2\}$	Re-calculating commitment
... $\{\mathcal{H}(m_1, n_1), k^+\}$	Verification ZK proof 1
... $n_3$	Commitment ZK proof 1
... $\text{cred}_{k^-}^{m_1}(m_2; \{n_2, n_5\})$	Recognise signature
... $\{k^+, m_2\}$	Verification ZK proof 2
... $n_6$	Commitment ZK proof 2

Table 4: Testability relation  $* \Rightarrow *$ :  $m, m_i, n, n_i$  context messages;  $k^+, k^-$  context private/public keys. The second column describes which test that actor(s) can perform.

CL signatures [8]. An anonymous credential is usually derived by the user from the transcript of its issuing protocol ( $\vdash\text{EI}_3$ ) (the issuer does not know  $n_2$  and so does not learn the credential); but it can also be constructed directly from its components ( $\vdash\text{CR}$ ). Similarly for the issuing protocol transcript itself ( $\vdash\text{CI}$ ). Before the issuing protocol takes place, the user needs to have sent a randomised commitment  $\mathcal{H}(m_1, n_1)$  to her secret identifier to the issuer. During the protocol, additional randomness  $n_2, \dots, n_8$  is generated by the two parties;  $n_1, \dots, n_8$  together form the randomness component of the ICred primitive. Inference rules ( $\vdash\text{EI}_1$ ) and ( $\vdash\text{EI}_2$ ) model the inference of secret information from the transcript using randomness and other inputs to the protocol. As with our model of ZK proofs, we only consider rules needed to infer personal information, and assume non-re-use of randomness. In Appendix A.2 we explain why these rules accurately capture privacy aspects.

The testing rule ( $\vdash\text{T}$ ) (Figure 7) expresses that for certain primitives, actors can test whether specific inputs were used. For instance, an actor can try to verify a signature  $S_{k^-}(m)$  using any known public key  $k'^+$  and message  $m'$ ; if verification succeeds, he learns that  $k^+$  has the same contents as  $k'^+$ , and  $m$  has the same contents as  $m'$ . We call  $\{k^+, m\}$  *testable* from  $S_{k^-}(m)$ , denoted  $S_{k^-}(m) \Rightarrow \{k^+, m\}$ . If  $n$  is testable from  $m$  and an actor can deduce a message  $n'$  with the same contents as  $n$ , then by ( $\vdash\text{T}$ ) he can also deduce  $n$ .

The testability relation  $\Rightarrow$  is defined in Table 4. Symmetric and (labelled) asymmetric encryptions allow for testing of the decryption key.<sup>1</sup> Signatures allow for signature verification. Similarly, an anonymous credential can be verified to correspond to a given verification key, message and secret identifier. From a ZK proof, the public information and user's randomness can be tested (Appendix A.1). A credential issuing protocol transcript allows for testing of various nonces and information used (Appendix A.2); in particular, all messages needed for the elimination rules ( $\vdash\text{EI}_1$ )–( $\vdash\text{EI}_3$ ) are testable.

**Example 2.** Let  $\mathcal{C}_a = \{E'_k(da)|^\pi, l|^\rho\}$  be the set of messages known by an actor  $a$ , with  $k|^\pi \doteq l|^\rho$ . Then  $\mathcal{C}_a \vdash da|^\pi$  can be deduced as follows:

<sup>1</sup>This is an over-estimation in case the plaintext is unknown, random, and unformatted, e.g., a nonce.

$$\begin{array}{c}
\frac{}{\mathcal{C}_a \vdash id|_2^\eta} (\vdash \mathbf{0}) \quad \frac{}{\mathcal{C}_a \vdash age|_3^\eta} (\vdash \mathbf{0}) \\
\hline
\mathcal{C}_a \vdash \{id|_2^\eta, age|_3^\eta\} \quad (\vdash \mathbf{CC}) \\
\frac{}{\mathcal{C}_a \vdash id|_2^\eta} (\vdash \mathbf{0}) \quad \frac{\mathcal{C}_a \vdash \{id|_2^\eta, age|_3^\eta\}}{\mathcal{C}_a \vdash \mathcal{H}(id|_2^\eta, age|_3^\eta)} (\vdash \mathbf{CH}) \quad \frac{}{\mathcal{C}_a \vdash \mathcal{H}(id, age)|_1^\eta} (\vdash \mathbf{0}) \\
\hline
\mathcal{C}_a \vdash id|_1^\eta \quad (\vdash \mathbf{C})
\end{array}$$

Figure 11: Deduction of  $id|_1^\eta$  given message set  $\mathcal{C}_a = \{\mathcal{H}(id, age)|_1^\eta, id|_2^\eta, age|_3^\eta\}$  (Example 5).

$$\begin{array}{c}
\frac{}{\mathcal{C}_a \vdash E'_k(da)|^\pi} (\vdash \mathbf{0}) \quad \frac{}{\mathcal{C}_a \vdash E'_k(da)|^\pi} (\vdash \mathbf{0}) \quad \frac{}{\mathcal{C}_a \vdash l|^\rho} (\vdash \mathbf{0}) \\
\hline
\mathcal{C}_a \vdash E'_k(da)|^\pi \quad \mathcal{C}_a \vdash l|^\rho \quad (\vdash \mathbf{T}) \\
\hline
\mathcal{C}_a \vdash k|^\pi \quad (\vdash \mathbf{DE}) \\
\hline
\mathcal{C}_a \vdash da|^\pi
\end{array}$$

The deduction models an actor(s) testing whether  $l|^\rho$  is the decryption key for  $E'_k(da)|^\pi$  ( $\vdash \mathbf{T}$ ). By learning it, the actor(s) can decrypt the message ( $\vdash \mathbf{DE}$ ).  $\square$

The final rule of our deductive system is the content analysis rule ( $\vdash \mathbf{C}$ ) (Figure 8). It models conclusions an actor can draw from seeing messages with the same contents in different contexts. The statement of the rule relies on the syntactic structure of messages, which we first elaborate on.

The syntactic structure of messages describes how they are constructed from data items, identifiers, and non-personal information using cryptographic primitives. The components of these primitives are numbered according to the order defined by the corresponding construction rules in Figures 8, 9, and 10 : e.g.,  $E'_n(m)$  has first component  $m$  and second component  $n$ . (For the AKA primitive, we take the private keys and nonces as submessages.) Recursively, submessages  $n$  of  $m$  have a well-defined “position”  $z$  in  $m$ , and we write  $n = m@z$ .

**Example 3.** The message  $\mathcal{H}(E'_n(m))$  has four submessages:

(i)  $\mathcal{H}(E'_n(m))@1 = \mathcal{H}(E'_n(m))$ ; (ii)  $\mathcal{H}(E'_n(m))@2 = E'_n(m)$ ; (iii)  $\mathcal{H}(E'_n(m))@3 = m$ ; and (iv)  $\mathcal{H}(E'_n(m))@4 = n$ .  $\square$

If two context messages  $m_1$  and  $m_2$  are content equivalent, then the properties of  $\sigma$  and  $\tau$  imply content equivalence of their submessages. That is, if  $m_1@z$  and  $m_2@z$  are defined (i.e., there exists a submessage at position  $z$ ), then they are content equivalent. In addition, if  $m_1@z = k^+$  and  $m_2@z = k'^+$ , then not only  $k^+ \doteq k'^+$  follows, but also  $k^- \doteq k'^-$ , and vice versa. Similarly, if  $m_1$  and  $m_2$  contain data items satisfying a property  $\psi_k$ , the content equivalence of that property is also implied. Formally:

**Definition 4.** Let  $m_1, m_2, n_1, n_2$  be context messages. The pair  $(m_1, m_2)$  is *evidence* for  $n_1 \doteq n_2$ , denoted  $(m_1 \doteq m_2) \Rightarrow (n_1 \doteq n_2)$ , if  $m_1 \doteq m_2$ , and for some  $z, k$  and all  $i \in \{1, 2\}$ , one of the following three conditions holds:

1.  $n_i = m_i@z$ ;
2.  $m_i@z$  and  $n_i$  form a private/public key pair;
3.  $n_i$  is a property  $\psi_k$  of  $m_i@z$ .

The “content analysis” inference rule ( $\vdash\text{C}$ ) then states that if an actor can derive evidence  $(m_1, m_2)$  for  $n_1 \doteq n_2$  and he can derive a message with  $n_1$  in it, then he can also derive the message with  $n_1$  replaced by  $n_2$ , and vice versa.

**Example 5.** Let  $\mathcal{C}_a = \{\mathcal{H}(id, age)|_1^\eta, id|_2^\eta, age|_3^\eta\}$  be the set of messages known by actor  $a$  with  $id \in I$ ,  $age \in D$  such that  $id|_1^\eta \doteq id|_2^\eta$  and  $age|_1^\eta \doteq age|_3^\eta$ .  $\mathcal{C}_a \vdash \mathcal{H}(id, age)|_1^\eta$  holds, and by ( $\vdash\text{CC}$ ), ( $\vdash\text{CH}$ ) we have  $\mathcal{C}_a \vdash \mathcal{H}(id|_2^\eta, age|_3^\eta)$ . From this,  $a$  knows that  $id|_1^\eta \doteq id|_2^\eta$  (as well as  $age|_1^\eta \doteq age|_3^\eta$ ). By ( $\vdash\text{C}$ ) he can then deduce  $id|_1^\eta$  (Figure 11). In the same way also  $\mathcal{C}_a \vdash age|_1^\eta$  follows.  $\square$

In [71], we formally compared the expressiveness of our context-layer deductive system to that of the corresponding standard information-layer deductive system. We showed that for any piece of information, there is at least one context-layer representation that our system can derive. Therefore, no information is lost. The deductive system presented here is an extension to the one presented in [71], and the same line of reasoning applies here as well. Intuitively, for every elimination rule, testing rules can be applied to derive context messages that satisfy the prerequisites for the rule. Thus, one elimination rule at the information layer corresponds to one elimination rule and some testing rules at the object layer.

### 3.2.3 Associability

After detectability, we now consider the other type of knowledge held by an actor: associability. For associability, in addition to messages, the set  $\mathcal{C}_a$  contains the context entities that  $a$  knows:  $\mathcal{C}_a \subset \mathcal{L}^c \cup \mathcal{E}^c$ . Associations between context items follow from properties of both  $\sigma$  and  $\tau$ . First, context items in one context are related, and so is the same entity in different contexts (properties 3, 4 of  $\sigma$ ). Second, context identifiers with equal contents are equal (property 1 of  $\tau$ ). Thus, define the *associability relation*  $\leftrightarrow_a$  of actor(s)  $a$  as the minimal equivalence relation on  $\mathcal{O}^c$  such that:

1. For all  $e|_k^\eta, e|_l^\zeta \in \mathcal{C}_a \cap \mathcal{E}^c$ :  $e|_k^\eta \leftrightarrow_a e|_l^\zeta$ ;
2. For all  $x|_k^\eta, y|_k^\eta \in \mathcal{O}^c$ :  $x|_k^\eta \leftrightarrow_a y|_k^\eta$ ;
3. If  $\mathcal{C}_a \vdash m_1$ ,  $\mathcal{C}_a \vdash m_2$ , and  $(m_1 \doteq m_2) \Rightarrow (i_1 \doteq i_2)$  for  $i_1, i_2 \in I^c$ , then  $i_1 \leftrightarrow_a i_2$ .

For associability by a coalition  $\{a_1, \dots, a_n\} \subset \mathcal{A}$ , written as  $\leftrightarrow_{a_1 + \dots + a_n}$ , the above rules are applied to  $\mathcal{C}_A = \mathcal{C}_{a_1} \cup \dots \cup \mathcal{C}_{a_n}$ .

Note that actors may associate items which they cannot detect. In fact, because of transitivity of  $\leftrightarrow_a$ , this may help to establish a relation between detectable items:

**Example 6.** Let  $\mathcal{C}_a = \{\{E_{shakey}|_1(id|_1), d|_1\}^\eta, \{E_{shakey}|_1(id|_1), d'|_1\}^\chi\}$  be the set of messages known by an actor  $a$ , where  $E_{shakey}|_1(id|_1)^\eta \doteq E_{shakey}|_1(id|_1)^\chi$ . Then,  $id|_1^\eta \leftrightarrow_a id|_1^\chi$  by condition 2 for  $\leftrightarrow_a$  (even though the actor can detect neither context identifier). By condition 1 for  $\leftrightarrow_a$  and transitivity,  $d|_1^\eta \leftrightarrow_a d'|_1^\chi$  follows.  $\square$

## 3.3 Message transmission

Actors increase their knowledge of personal information by exchanging messages. We model the increase of knowledge in a particular system evolution by a sequence of *states*. A state comprises the sets  $\{\mathcal{C}_x^t\}_{x \in \mathcal{A}}$  of messages and entities known by each actor

$x \in \mathcal{A}$  at time  $t = 0, 1, \dots, n$ . The three-layer model is defined independently from the state, and contains all items and messages within the system (including items generated during the protocol execution, such as nonces). For each actor (or coalition of actors)  $a$ , the state  $\{\mathcal{C}_x^t\}_{x \in \mathcal{A}}$  determines detectability  $\mathcal{C}_a^t \vdash \dots$  as defined by the deductive system in Section 3.2.2, and associability  $\leftrightarrow_a$  as defined in Section 3.2.3 (we write  $\leftrightarrow_a^t$  if we want to make the state explicit).

The transition from one state to another corresponds to a message transmission. A message transmission involves two parties; each party is modelled by an identifier representing its communication address in the transmission. The simplest type of message transmission is one actor sending a message to another; two other types model the execution of cryptographic protocols.

**Definition 7.** A message transmission can be of three types:

1.  $a \rightarrow b : m$ ;
2.  $a \mapsto b : \text{ZK}(m_1; m_2; m_3; m_4)$ ;
3.  $a \mapsto b : \text{ICred}_{k^-}^{m_1}(m_2; n)$ ,

with  $a, b$  context identifiers;  $k^- \in K_c$  a context private key, and  $m, m_i, n$  context messages. A trace  $\mathfrak{T}$  is a sequence of message transmissions  $t_i$ , denoted  $\mathfrak{T} = t_1; t_2; \dots; t_n$ .

In message transmission  $a \mapsto b : \text{ZK}(\dots)$ ,  $a$  is the prover and  $b$  the verifier; in  $a \mapsto b : \text{ICred}(\dots)$ ,  $a$  is the user and  $b$  the issuer.

To verify the validity of a message transmission, it is necessary to verify whether an actor  $a$  can send a message  $m$  in a certain state  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$ . He can certainly do so if he can derive the message from his knowledge base ( $\mathcal{C}_a^t \vdash m$ ); however, it is also possible that  $m$  (or its submessages) has not been observed by any actor in that state, so  $\mathcal{C}_a \vdash m$  does not hold (hereafter we call such messages *undetermined*). In this case,  $a$  needs to “instantiate”  $m$  by deriving a message equivalent to  $m$  from the messages he knows. Note that at that point in time,  $a$  may have different choices about what information to send. However, we are only interested in the choice determined by  $\sigma$  corresponding to the system evolution we consider, and not in any alternative choices leading to alternative system evolutions. We first formalise the notion of “determined”.

**Definition 8.** Let  $\{\mathcal{C}_x^t\}_{x \in \mathcal{A}}$  be a state at time  $t$ .

- Context item  $n$  is *determined* in state  $\{\mathcal{C}_x^t\}_{x \in \mathcal{A}}$  if there is an actor  $a \in \mathcal{A}$ , message  $m \in \mathcal{C}_a$ , and  $z$  such that: (1)  $n = m@z$ ; or (2)  $n$  and  $m@z$  form a private/public key pair; or (3)  $n$  is a property of  $m@z$ .
- Context message  $n$  is *determined* in a state if all context items occurring in  $n$  are determined.
- Context  $(\eta, k)$  is *determined* in a state if at least one context item  $x_k^\eta$  in  $(\eta, k)$  is determined (i.e., the data subject of the context is defined).

The following definition formalises the intuition that to send an undetermined message  $m$ ,  $a$  should derive an equivalent message  $n$  in which all undetermined items are “instantiated”:

**Definition 9.** Let  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  be a state at time  $t$ . A message  $m$  is *determinable* by actor  $a$  in  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  if  $\mathcal{C}_a^t \vdash n$  for some  $n \equiv m$  satisfying the following conditions:

$t =$	Determinable by $a$	Determinable by $b$
$a \rightarrow b : m$	$\{a, b, m\}$	$\emptyset$
$a \mapsto b : \text{ZK}(m_1; m_2; m_3; \{n_a, n_b\})$	$\{a, b, m_1, n_a\}$	$n_b$
$a \mapsto b : \text{ICred}_k^{m_1}(m_2; \{n_i\}_{i=1}^7)$	$\{a, b, k^+, m_1, n_1, n_2, n_3, n_7\}$	$\{k^+, k^-, m_2, n_4, n_5, n_6\}$

Table 5: Determinability requirements for the different types of message transmissions

1. Whenever  $m@z$  is determined, then  $m@z = n@z$ .
2. Whenever  $m@z_1 = m@z_2$ , then  $n@z_1 = n@z_2$ .
3. If  $m@z$  is a context item whose context  $(\eta, k)$  is determined in  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  with  $k \neq \cdot$ , then  $n@z \leftrightarrow_a *|_k^\eta$  where  $*|_u^\pi$  represents any message associable to  $(\pi, u)$ .
4. If  $m@z_1, m@z_2$  are context items whose context  $(\eta, k)$  is not determined in  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  with  $k \neq \cdot$ , then  $n@z_1 \leftrightarrow_a n@z_2$ .

The message  $n$  chosen as instantiation for  $m$  needs to be equivalent to  $m$ , i.e., have the same structure as  $m$  and contain the same information; it also needs to satisfy the additional constraints from conditions 1–4. Condition 1 states that submessages of  $m$  that are already determined should be the same in  $n$ . Condition 2 states that when the same item occurs multiple times in  $m$ , then all occurrences should have the same instantiation in  $n$ . By condition 3, if  $m$  contains an undetermined item  $m@z$  whose context is determined, the actor should be able to associate the instantiation of  $m@z$  to that context. Similarly, condition 4 requires that if  $m$  contains more than one item from the same context, then the actor should be able to associate their instantiations. Note that if  $m$  is determined, then determinability is the same as derivability.

**Example 10.** Let  $m = \{id, age\}|_u^\pi$  be the message to be sent by an actor  $a$  in a state in which  $id|_u^\pi$  is determined and  $age|_u^\pi$  is undetermined. Then,  $m$  is determinable if  $a$  can derive a message  $\{id|_u^\pi, *\}$  such that  $* \equiv age|_u^\pi$  and  $id|_u^\pi \leftrightarrow_a *$ .  $\square$

We now define when message transmissions and traces are *valid*, i.e., they can be executed. For message transmissions of type 1, validity means determinability by the sender of the message and of the sender and receiver identifiers. For the other types, the initiator of the protocol should determine the sender and receiver addresses, and both parties contribute to the construction of the message to be transmitted. A transmission results in a new state in which both parties know the protocol transcript. This intuition can be extended to traces.

**Definition 11.** Let  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  be a state at time  $t$ . Let  $a$  denote the entity corresponding to context identifier  $a$  (i.e.,  $a \leftrightarrow a$ ), and  $b$  the entity corresponding to  $b$  (i.e.,  $b \leftrightarrow b$ ).

- The message transmission  $t$  is *valid* in  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  if  $a$  and  $b$  can determine the messages indicated in Table 5.
- The *resulting state* of message transmission  $t = a \rightarrow b : m$  or  $t = a \mapsto b : m$  from  $\{\mathcal{C}_a^t\}_{a \in \mathcal{A}}$  is the state  $\{\mathcal{C}_a^{t+1}\}_{a \in \mathcal{A}}$  such that  $\mathcal{C}_a^{t+1} = \mathcal{C}_a^t \cup \{a, b, m\}$ ,  $\mathcal{C}_b^{t+1} = \mathcal{C}_b^t \cup \{a, b, m\}$ .
- The validity and resulting state of a trace  $\mathcal{T} = t_1; \dots; t_n$  are defined iteratively.

For ZK proofs, the prover needs to know the private information for the proof, and both parties contribute randomness. Note that to participate in the protocol, the verifier

```

1: {Let  $\models$  denote the deductive system without the content analysis rule}
2: for all context items  $m'$ :  $m' \doteq m, C_a \models m'$  do
3:   for all context items  $p, p'$ :  $m@z = p, m'@z = p', p \neq p'$  do
4:     {Find sequence of evidence for  $p \doteq p'$  using breadth-first search}
5:      $Q \leftarrow \{p\}$  {queue of items to check};  $P \leftarrow \{\}$  {already checked}; found  $\leftarrow$  false
6:     while  $Q \neq \{\} \wedge \neg \text{found}$  do
7:        $q \leftarrow \text{pop}(Q); P \leftarrow P \cup \{q\}$  {move  $q$  from queue to already checked}
8:       if  $q = p'$  then found  $\leftarrow$  true; break {evidence for  $p \doteq p'$  found} end if
9:       for all context items  $q'$ :  $q'$  occurs in message in  $C_a, q' \doteq q, q' \notin P \cup Q$  do
10:        {Try to find evidence for  $q \doteq q'$ }
11:        for all context items  $n$ :  $C_a \models n, n$  is minimal w.r.t.  $q$  do
12:          if  $\exists n' : C_a \models n' : (n \doteq n') \Rightarrow (q \doteq q')$  then  $Q \leftarrow Q \cup \{q'\}$  end if
13:        end for
14:        for all context items  $n'$ :  $C_a \models n', n'$  is minimal w.r.t.  $q'$  do
15:          if  $\exists n : C_a \models n : (n \doteq n') \Rightarrow (q \doteq q')$  then  $Q \leftarrow Q \cup \{q'\}$  end if
16:        end for
17:      end for
18:    end while
19:    if  $\neg \text{found}$  then break {No such  $p'$  found: try next  $m'$ } end if
20:  end for
21:  return true {Actor has evidence that  $m \doteq m'$  for a  $m'$  such that  $C_a \models m'$ }
22: end for
23: return false {For all  $m'$  such that  $m \doteq m', C_a \models m'$ : actor has no evidence for  $m \doteq m'$ }

```

Figure 12: Algorithm implementing the deductive system: given message set  $C_a$  and context message  $m$ , check whether  $C_a \vdash m$

does not need to know the public information or the properties to be proven; however, he does need to know this information to be able to interpret the proof (i.e., to apply the testing rule). For credential issuing, the user needs to know her secret identifier  $m_1$ , randomness, and the issuer's public key; the issuer needs to know his private/public key pair, the attributes to be signed, and additional randomness.

**Example 12.** Consider actors  $\mathcal{A} = \{gov, store\}$  communicating using addresses  $ip|_{gov}^\cdot, ip|_{store}^\cdot$ . Actor *gov* knows its own address, and holds a database containing user information:  $C_{gov}^0 = \{ip|_{gov}^\cdot\} \cup \{id|_{33}^{db}, age|_{33}^{db}, \dots\}$ . Actor *store* knows a user identifier and the addresses of itself and *gov*:  $C_{store}^0 = \{ip|_{store}^\cdot, ip|_{gov}^\cdot, id|_{user}^\cdot\}$ . An instance  $\pi$  of a simple protocol in which *gov* and *store* exchange information about the user  $u$  (*store* is the client *cli*, and *gov* is the server *srv*) is modelled by the following valid trace:

$$ip|_{cli}^\pi \rightarrow ip|_{srv}^\pi : id|_u^\pi \ ; \ ip|_{srv}^\pi \rightarrow ip|_{cli}^\pi : \{id|_u^\pi, age|_u^\pi\}.$$

where  $ip|_{cli}^\pi \equiv ip|_{store}^\cdot, ip|_{srv}^\pi \equiv ip|_{gov}^\cdot, id|_u^\pi \equiv id|_{33}^{db} \equiv id|_{user}^\cdot$ , and  $age|_u^\pi \equiv age|_{33}^{db}$ . The resulting state from this trace is  $\{C_x^2\}_{x \in \mathcal{A}}$ . In this state,  $C_{store}^2 \vdash age|_u^\pi$  and  $id|_{user}^\cdot \leftrightarrow_a^2 age|_u^\pi$ .  $\square$

### 3.4 Prolog Implementation

We have implemented the framework presented in this section using Prolog. Here we describe our implementation<sup>2</sup> and its efficiency in general terms; for details, refer

<sup>2</sup>The implementation, along with its documentation, can be downloaded at <http://www.mobiman.me/publications/downloads/>.



to the documentation of the implementation.

### 3.4.1 Deductive system

Our deductive system is a traditional deductive system [29, 37] with two extra rules: testing and content analysis. Let us first ignore content analysis, and only consider the construction, testing and elimination rules. Construction rules generally derive messages from submessages; testing and elimination rules derive submessages from messages using “additional prerequisites” (e.g., the key for the decryption rule ( $\vdash\text{EE}$ )). As testing/elimination and construction cancel each other out, there is no point in applying testing/elimination to the result of construction rule. Thus, to check the derivability of a message  $m$ , we try to find a message  $n$  in which it occurs as submessage, and try to derive  $m$  from it using elimination and testing. If this does not work, we repeat the procedure for  $m$ ’s submessages: if successful, then  $m$  can be obtained from them with a construction rule.

While trying elimination or testing rules, we need to check the derivability of the additional prerequisites  $n$ . We claim that this check can be done at the contents layer (so a simple deductive system suffices). For the testing rule this is clear; however, it also holds for elimination rules because their additional prerequisites can always be obtained from a content equivalent message using the testing rule.

Thus, in terms of evaluation, our deductive system differs from standard systems in two ways. First, for elimination rules, the additional prerequisites are evaluated not using the deductive system itself, but using a (standard) deductive system at the contents layer. Second, testing rules are added which are evaluated in the same way as elimination rules. Intuitively, our deductive system is thus not much harder to evaluate than a corresponding standard deductive system. (However, typically it will be run on a larger message set because information has multiple representations.)

To implement the full deductive system, we use that any deduction in the full deductive system can be transformed into a deduction deriving the same message that satisfies:

- After content analysis rules, no other rules are applied to a message
- In any application of ( $\vdash\text{C}$ ), the message  $n_2$  and the message  $n_1$  from which it is derived only differ by one context item at one position
- In any application of ( $\vdash\text{C}$ ), the messages  $m_1$  and  $m_2$  are derived without content analysis; also,  $m_1$  is minimal with respect to  $n_1$  in the sense that no elimination or testing rule can be applied to it to obtain a submessage containing  $n_1$ ; and/or  $n_2$  is minimal with respect to  $m_2$ .

The algorithm in Figure 12 is an imperative translation of our Prolog implementation; by the above properties, it implements derivability in our full deductive system. Namely, to derive  $m$  from a given message set  $\mathcal{C}_a$ , it takes all messages  $m' \doteq m$  such that  $\mathcal{C}_a \vdash m'$ , and tries to obtain  $m'$  from  $m$  by content analysis in a context-item-by-context-item fashion. For all positions  $z$  at which  $m$  and  $m'$  differ, the algorithm performs a breadth-first search for messages obtained from  $m$  by content analysis at position  $z$ , until it finds  $m$  with  $m@z$  replaced by  $m'@z$ . The breadth-first search is performed by first searching for a minimal message using testing and elimination rules (lines 10 and 13); and then searching for a content equivalent message using testing, elimination and construction rules (lines 11 and 14). We did not optimise this algorithm in terms of

complexity. Indeed, in practice, most context items are content equivalent only to few other items, so the search space for the algorithm is very limited.

### 3.4.2 Associability and Trace Validity

The algorithm for checking the associability of two contexts is similar to the previous algorithm. In particular, it starts with one context  $(\eta, k)$  and uses breadth-first search to find associable contexts. This involves finding all identifiers (or entities) that occur in  $(\eta, k)$  and all other contexts in which that identifier occurs. The algorithm then searches evidence for content equivalence of the different representations of the identifier.

The main task in implementing traces is to check for determinability of a message  $m$ ; that is, to find a derivable message  $n$  that is equivalent to  $m$  and satisfies properties (1) to (3) from Definition 9. Properties (1) and (2) place restrictions on the form of the message, which can be expressed in terms of free variables in a Prolog query to the deductive system. For property (3) we check associability as above.

## 4 Privacy Analysis of IdM Systems

In this section we apply our method to analyse the IdM systems presented in Section 2.4. First, we model the actors and personal information in the case study presented in Section 2.3, describe who knows what initially, and translate the requirements in Table 1 to formal, verifiable properties (Section 4.1). Then, we formalise the IdM systems (Section 4.2). Finally, we verify whether the IdM systems satisfy the requirements and discuss the results (Section 4.3).

### 4.1 Formalizing the Case Study and Requirements

Figure 13 shows the actors in the case study and the contexts they occur in, together with the initial knowledge held by each of them. The actors are listed in Figure 13(a). (In this case study, actors and entities are the same.) The trusted third party  $ttp$  is included because of the anonymity revocation requirement; however, note that it only occurs in the Identity Mixer and Smartcard schemes.

Figure 13(b) lists the domains we use. The  $\cdot$  domain contains publicly known identifiers for the identity and service providers, and private/public key pairs. The  $\iota$ ,  $\kappa$ , and  $\lambda$  domains represent databases of user information held by the respective parties. The  $\pi$ ,  $\eta$ ,  $\zeta$ , and  $\xi$  domains represent the communication protocols that are executed during the case study. We consider two separate service provisions: this is needed to analyse session unlinkability. For simplicity, all communication related to one service provision is modeled in a single domain. This expresses that parties involved in service provision without communicating directly are able to link their views of the protocol. Alternatively, each pair of communication partners could have a separate domain.

Figure 13(c) shows the profiles representing the actors in the different domains. For instance, in the  $\cdot$ ,  $\iota$ ,  $\kappa$  and  $\lambda$  domains, Alice represented by the  $al$  profile; in the  $\pi$ ,  $\eta$ ,  $\zeta$ , and  $\xi$  domains, she is represented by  $u$ . By naming these profiles differently, we emphasise that actors learn the information not as information about Alice, but as information about “the purchaser in transaction  $x$ ”, etc.

Figures 13(d) and 13(e) define the initial state  $\{\mathcal{C}_a^s\}_{a \in \mathcal{A}}$  of the case study at the context layer. The information-layer and contents-layer descriptions result as follows.

$e \in \mathcal{E}$	Actor/entity	Dom.	Description	Entity	Domains
$al$	Alice	$\cdot$	identifiers/keys	$al$	$\cdot, \iota, \kappa, \mu$
$ii$	Address provider	$\iota$	Alice's knowledge	$ii$	$u$
$is$	Subscription provider	$\kappa$	$ii$ 's user database	$is$	$idp1$
$bs$	E-book store	$\mu$	$is$ 's user database	$bs$	$idp2$
$ttp$	Trusted third party	$\pi$	registration at $ii$	$ttp$	$sp$
		$\eta$	registration at $is$		$ttp$
		$\zeta, \xi$	service provisions		

(a) Actors/entities

(b) Domains

(c) Profiles

(d) Information about  $ii$ ,  $is$ ,  $bs$ ,  $ttp$ : anybody knows identifiers and public keys; actor knows own private key

Var	$al$	$ii$	$is$	$bs$	Description
$al$	$\{al _{al}^{\iota}, al _{u}^{\kappa}, al _{u}^{\mu}\}$	-	-	-	Entity
$i_{ii}$	$\{i_{ii} _{al}^{\iota}, i_{ii} _{u}^{\kappa}\}$	$\{i_{ii} _{al}^{\kappa}, i_{ii} _{u}^{\pi}\}$	-	-	Identifier at $ii$
$d_1$	$d_1 _{al}^{\iota}$	$d_1 _{al}^{\kappa}$	-	-	City
$d_2$	$d_2 _{al}^{\iota}$	$d_2 _{al}^{\kappa}$	-	-	Age
$d_2 > 60$	$d_2 > 60 _{al}^{\iota}$	$d_2 > 60 _{al}^{\kappa}$	-	-	age "> 60" property
$d_3$	$d_3 _{al}^{\iota}$	$d_3 _{al}^{\kappa}$	-	-	Address
$i_{is}$	$\{i_{is} _{al}^{\iota}, i_{is} _{u}^{\eta}\}$	-	$\{i_{is} _{al}^{\mu}, i_{is} _{u}^{\eta}\}$	-	Identifier at $is$
$d_5$	$d_5 _{al}^{\iota}$	-	$d_5 _{al}^{\mu}$	-	Subscription date
$d_6$	$d_6 _{al}^{\iota}$	-	$d_6 _{al}^{\mu}$	-	Subscription type
$d_7$	-	-	-	$\{d_7 _{u}^{\zeta}, d_7 _{u}^{\xi}\}$	Transaction details
$ip$	$\{ip _{u}^{\pi}, ip _{u}^{\eta}, ip _{u}^{\zeta}, ip _{u}^{\xi}\}$	-	-	-	IP address

(e) Information about Alice known initially by the different actors: rows represent types of information about Alice; the columns labelled  $al$ ,  $ii$ ,  $is$ , and  $bs$  represent the initial knowledge held by the respective actors

Figure 13: Formalisation of the case study

When context items about the same entity using the same variable are denoted in the normal font (e.g.  $i_{ii}|_{u}^{\pi}$  and  $i_{ii}|_{al}^{\kappa}$ ), they are equivalent; when denoted in boldface (e.g.  $ip|_{u}^{\pi}$ ,  $ip|_{u}^{\eta}$ ), they are all pairwise non-equivalent. Variables of the form  $i$ ,  $i_*$ ,  $k_*$ ,  $k_*$ ,  $ip$  (for any  $*$ ) denote identifiers; variables  $d_*$  denote data items; other variables denote non-personal information. All representations of a single information item use the same variable. Because this case study includes only one data subject, all pieces of information have unique contents, i.e., the information and contents layers coincide.

Figure 13(d) defines the information available about  $ii$ ,  $is$ , and  $bs$ . This information consists of a private/public key pair for each of the actors, and an identifier for  $ii$ ,  $is$ , and  $bs$ . The public keys and identifiers are known by everybody; each actor also knows his own private key.

Figure 13(e) lists the personal information known initially about Alice, grouped by variable. For instance,  $d_1$  represents a city; Alice knows her city as  $d_1|_{al}^{\iota}$  and  $ii$  knows it as  $d_1|_{al}^{\kappa}$ . We assume that the actual attribute exchange between user and identity provider has already taken place, as shown in the  $\kappa$  and  $\mu$  domains. Knowledge about Alice is also present in the  $\pi$ ,  $\eta$ ,  $\zeta$  and  $\xi$  domains representing protocols. Knowledge of  $i_{ii}|_{u}^{\pi}$ ,  $i_{is}|_{u}^{\eta}$  held by Alice and the respective identity providers represents the fact that Alice has authenticated to them. In the context of the two service provisions, Alice knows that she is the data subject ( $al|_{u}^{\zeta}$ ,  $al|_{u}^{\xi}$ ); the service provider knows transaction details  $d_7|_{u}^{\zeta}$ ,  $d_7|_{u}^{\xi}$ . Alice knows her IP address  $ip|_{u}^*$ ,  $*$   $\in \{\pi, \eta, \zeta, \xi\}$ ; note that it is assumed to change dynamically between sessions.

In Table 6 we formalise the requirements from Table 1 with respect to the case

Requirement	Formalisation
Attribute exchange (AX)	$\mathcal{C}_{bs} \vdash \{d_1, d_2 > 60, d_6\}^{\zeta}_{ u}, \{d_1, d_2 > 60, d_6\}^{\xi}_{ u}$
Anonymity revocation (AR)	$* _{al}^{\kappa} \leftrightarrow_{bs+ii+is+tp} * _u^{\zeta} \leftrightarrow_{bs+ii+is+tp} * _u^{\xi}$
Irrelevant attribute undetectability (SID)	$\mathcal{C}_{bs} \not\vdash d_3 _*^* \wedge \mathcal{C}_{bs} \not\vdash d_5 _*^*$
Property-attribute undetectability (SPD)	$\mathcal{C}_{bs} \not\vdash d_2 _*^*$
IdP attribute undetectability (ID)	$\mathcal{C}_{is} \not\vdash d_1 _*^* \wedge \mathcal{C}_{is} \not\vdash d_2 _*^* \wedge \mathcal{C}_{is} \not\vdash d_3 _*^* \wedge$ $\mathcal{C}_{is} \not\vdash d_2 > 60 _*^* \wedge \mathcal{C}_{ii} \not\vdash d_5 _*^* \wedge \mathcal{C}_{ii} \not\vdash d_6 _*^*$
Mutual IdP involvement undetectability (IM)	$\neg(\exists p : * _{is}^{\zeta} \leftrightarrow_{ii} * _{idp2}^p \wedge * _u^p \leftrightarrow_{ii} * _{al}^{\kappa}) \wedge$ $\neg(\exists p : * _{ii}^{\zeta} \leftrightarrow_{is} * _{idp1}^p \wedge * _u^p \leftrightarrow_{is} * _{al}^{\mu})$
IdP-SP involvement undetectability (ISM)	$\neg(\exists p : * _{bs}^{\zeta} \leftrightarrow_{ii} * _{sp}^p \wedge * _u^p \leftrightarrow_{ii} * _{al}^{\kappa}) \wedge$ $\neg(\exists p : * _{bs}^{\zeta} \leftrightarrow_{is} * _{sp}^p \wedge * _u^p \leftrightarrow_{is} * _{al}^{\mu})$
Session unlinkability (SL)	$* _u^{\zeta} \leftrightarrow_{bs} * _u^{\xi}$
IdP service access undetectability (IL)	$* _{al}^{\kappa} \leftrightarrow_{ii} * _u^{\zeta}) \wedge * _{al}^{\kappa} \leftrightarrow_{ii} * _u^{\xi} \wedge$ $* _{al}^{\mu} \leftrightarrow_{is} * _u^{\zeta} \wedge * _{al}^{\mu} \leftrightarrow_{is} * _u^{\xi}$
IdP profile unlinkability (IIL)	$* _{al}^{\kappa} \leftrightarrow_{ii+is} * _{al}^{\mu}$
IdP/SP unlinkability (ISL)	$* _{al}^{\kappa} \leftrightarrow_{ii+is+bs} * _u^{\zeta} \wedge * _{al}^{\mu} \leftrightarrow_{ii+is+bs} * _u^{\xi} \wedge$ $* _{al}^{\kappa} \leftrightarrow_{ii+is+bs} * _u^{\zeta} \wedge * _{al}^{\mu} \leftrightarrow_{ii+is+bs} * _u^{\xi}$

Table 6: Formalisation of requirements in our case study ( $\mathcal{C}_a \vdash m$  means  $\neg\mathcal{C}_a \vdash m$ ;  $m \leftrightarrow_a n$  means  $\neg(m \leftrightarrow_a n)$ ;  $*$  means for all possible values)

study. AX and AR translate to detectability and associability properties of elements in our model. For AX, note that  $bs$  can always associate the personal information of the user to the purchase because of the common context  $(\zeta, u)$  or  $(\xi, u)$ , so we do not check this. Undetectability requirements are straightforward to formalise; e.g., property-attribute undetectability means undetectability by  $bs$  of the context item  $d_2|_p^{\delta}$  in any context  $(\delta, p)$ . Involvement requirements are defined as not knowing that a service or identity provider has a profile in the same domain as the user; for instance, for IM, there should be no domain  $p$  in which  $ii$  can link the  $idp2$  profile to  $|_{idp2}^{\kappa}$  and the  $u$  profile to  $|_{al}^{\kappa}$ . Linkability requirements translate to contexts not being associable by an actor or coalition.

## 4.2 Formalizing the IdM systems

We now present the formalisation of the different systems presented in Section 2.4. In each case, the formalisation consists of an initial state and a trace. The initial state  $\{\mathcal{C}_a^0\}_{a \in \mathcal{A}}$  extends the initial knowledge described in Figure 13 with respect to the specific system. The trace Scenario consists of the messages transmitted during registration at  $ii$ , registration at  $is$ , and two service provisions at  $bs$ , respectively.

We introduce the abbreviation  $MS_{k^-}(m) := \{m, S_{k^-}(m)\}$  to denote a message along with its signature, capturing both X.509 certificates [44] and signed SAML assertions [25].

### 4.2.1 Smart Certificates

Figure 14 displays our formalisation of smart certificates (Section 2.4.1). The top part defines the initial state  $\mathcal{C}^0$ . In addition to the knowledge from Figure 13, Alice knows her public key certificate  $MS_{k^-|_{ca}}(i|_{al}, k^+|_u, n_c|_c)$  ( $n_c|_c$  represents additional information in the certificate such as the validity date), and the corresponding private key  $k^-|_{al}$ .

$$\begin{aligned}
\mathcal{C}_{al}^0 &= \mathcal{C}_{al}^s \cup \{\text{MS}_{k^-|_{ca}}(i|_{al}, k^+|_{u, n_c|}), k^-|_{al}, \mathbf{n}_{z, a}|^\pi, \mathbf{n}_{z, a}|^\eta, \mathbf{n}_{z, a}|^\zeta, \mathbf{n}_{z, a}|^\xi\}; \\
\mathcal{C}_{ii}^0 &= \mathcal{C}_{ii}^s \cup \{\mathbf{n}_{z, b}|^\pi, n_a|^\pi\}; \\
\mathcal{C}_{is}^0 &= \mathcal{C}_{is}^s \cup \{\mathbf{n}_{z, b}|^\eta, n_b|^\eta\}; \\
\mathcal{C}_{bs}^0 &= \mathcal{C}_{bs}^s \cup \{\mathbf{n}_{z, b}|^\zeta, \mathbf{n}_{z, b}|^\xi\}
\end{aligned}$$

$$\text{Scenario} := \text{Reg}_1|^\pi; \text{Reg}_2|^\eta; \text{ServProv}|^\zeta; \text{ServProv}|^\xi$$

$$\text{Reg}_1 :=$$

$$\mathbf{ip}|_u \rightarrow ip|_{idp1} : \text{MS}_{k^-|_{ca}}(i|_u, k^+|_{u, n_c|}); \quad (1)$$

$$\mathbf{ip}|_u \mapsto ip|_{idp1} : \text{ZK}(k^-|_u; k^+|_u; \emptyset; \mathbf{n}_{z, a}|, \mathbf{n}_{z, b}|); \quad (2)$$

$$ip|_{idp1} \rightarrow \mathbf{ip}|_u : \text{MS}_{k^-|_{idp1}}(i|_u, d_1|_u, d_2|_u, d_3|_u, n_a|); \quad (3)$$

$$\text{Reg}_2 :=$$

$$\mathbf{ip}|_u \rightarrow ip|_{idp2} : \text{MS}_{k^-|_{ca}}(i|_u, k^+|_{u, n_c|}); \quad (4)$$

$$\mathbf{ip}|_u \mapsto ip|_{idp2} : \text{ZK}(k^-|_u; k^+|_u; \emptyset; \mathbf{n}_{z, a}|, \mathbf{n}_{z, b}|); \quad (5)$$

$$ip|_{idp2} \rightarrow \mathbf{ip}|_u : \text{MS}_{k^-|_{idp2}}(i|_u^\eta, d_5|_u^\eta, d_6|_u^\eta, n_b|^\eta); \quad (6)$$

$$\text{ServProv} :=$$

$$\mathbf{ip}|_u \rightarrow ip|_{sp} : \text{MS}_{k^-|_{ca}}(i|_u, k^+|_{u, n_c|}), \text{MS}_{k^-|_{idp1}}(i|_u, d_1|_u, d_2|_u, d_3|_u, n_a|); \quad (1)$$

$$\text{MS}_{k^-|_{idp2}}(i|_u, d_5|_u, d_6|_u, n_b|);$$

$$\mathbf{ip}|_u \mapsto ip|_{sp} : \text{ZK}(k^-|_u; k^+|_u; \emptyset; \mathbf{n}_{z, a}|, \mathbf{n}_{z, b}|); \quad (2)$$

Figure 14: Formalisation of smart certificates: state  $\{\mathcal{C}_a^s\}_{a \in \mathcal{A}}$  and (valid) trace Scenario

The other items of initial knowledge are the contributions  $\mathbf{n}_{z, *}|^*$  to Alice's proof of knowledge of  $k^-|_{al}$ , and additional information  $n_a|^\pi, n_b|^\eta$  put in the attribute certificates issued by *ii* and *is*.

The bottom part of Figure 14 defines the trace Scenario modelling communication in smart certificates. The messages in the traces  $\text{Reg}_1$  and  $\text{Reg}_2$  correspond to those in Figure 2(a); the messages in the trace  $\text{ServProv}$  correspond to those in Figure 2(b). We model the proof that Alice knows the secret key corresponding to her public key as a ZK proof with secret information  $k^-|_u$  and public information  $k^+|_u$ .

#### 4.2.2 Linking Service Model

Figure 15 shows the formalisation of the linking service model (Section 2.4.2). The top part defines the initial state  $\mathcal{C}^0$ . This system introduces the linking service *ls* as an additional actor: it has an address and a private/public key pair. *ls* and *is* have publicly known identifiers  $i|_{ls}, i|_{is}$  used in the referrals. The user database of *ls*, modelled by domain  $v$ , contains an entry for the user containing only the identifier  $i_l|_{al}^v$ . User authentication to *ls* during registration is modelled by *ls*'s knowledge of  $i_l|_u^\pi$ ; the pseudonyms generated by the identity providers are modelled as  $i_{i1, ls}|_u^\pi$  and  $i_{i2, ls}|_u^\pi$ . Alice's authentication at *ii* during service provision is modelled by the fact that *ii* knows the identifiers  $i_{ii}|_u^*, * \in \{\zeta, \xi\}$ .

The bottom part of Figure 14 defines the trace Scenario modelling communication

$$\begin{aligned}
\mathcal{C}_{ii}^0 &= \mathcal{C}_{ii}^s \cup \{ip|_{ls}, k^+|_{ls}, i|_{ls}, i|_{is}, i_{i1,ls}|_u, \pi|_u, \pi|_u, i_{ii}|_u, \mathbf{isess}|_u, \mathbf{n}|_u, \mathbf{n}|_u, i_{ii}|_u, \mathbf{isess}|_u, \mathbf{n}|_u, \mathbf{n}|_u\}; \\
\mathcal{C}_{is}^0 &= \mathcal{C}_{is}^s \cup \{ip|_{ls}, k^+|_{ls}, i|_{ls}, i|_{is}, i_{i2,ls}|_u, \mathbf{n}|_u, \mathbf{n}|_u\}; \\
\mathcal{C}_{bs}^0 &= \mathcal{C}_{bs}^s \cup \{ip|_{ls}, k^+|_{ls}, i|_{ls}, i|_{is}\}; \\
\mathcal{C}_{ls}^0 &= \mathcal{C}_{ls}^s \cup \{ip|_{ls}, k^+|_{ls}, k^-|_{ls}, i|_{ls}, i|_{is}, i|_{al}, i|_{al}, i|_{al}, i|_{al}, i|_{al}, i|_{al}, \mathbf{n}|_u, \mathbf{n}|_u, \mathbf{n}|_u, \mathbf{n}|_u\}
\end{aligned}$$

Scenario := Reg<sub>1</sub>|<sup>π</sup>; Reg<sub>2</sub>|<sup>η</sup>; ServProv|<sup>ζ</sup>; ServProv|<sup>ξ</sup>

Reg<sub>1</sub> :=

$$ip|_{idp1} \rightarrow ip|_{ls} : \text{MS}_{k^-|_{idp1}}(i_{i1,ls}|_u, \mathbf{n}|_u) \quad (1)$$

Reg<sub>2</sub> :=

$$ip|_{idp2} \rightarrow ip|_{ls} : \text{MS}_{k^-|_{idp2}}(i_{i2,ls}|_u, \mathbf{n}|_u) \quad (2)$$

ServProv :=

$$ip|_{idp1} \rightarrow ip|_{sp} : \text{MS}_{k^-|_{idp1}}(\mathbf{isess}|_u, d_1|_u, d_2|_u, i|_{ls}, E_{k^+|_{ls}}(i_{i1,ls}|_u, \mathbf{n}|_u)) \quad (1)$$

$$ip|_{sp} \rightarrow ip|_{ls} : E_{k^+|_{ls}}(i_{i1,ls}|_u, \mathbf{n}|_u), \text{MS}_{k^-|_{idp1}}(\mathbf{isess}|_u, d_1|_u, d_2|_u, i|_{ls}, E_{k^+|_{ls}}(i_{i1,ls}|_u, \mathbf{n}|_u)); \quad (2)$$

$$ip|_{ls} \rightarrow ip|_{sp} : i|_{idp2}, E_{k^+|_{idp2}}(i_{i2,ls}|_u, \mathbf{n}'|_u); \quad (3)$$

$$ip|_{sp} \rightarrow ip|_{idp2} : E_{k^+|_{idp2}}(i_{i2,ls}|_u, \mathbf{n}'|_u), \text{MS}_{k^-|_{idp1}}(\mathbf{isess}|_u, d_1|_u, d_2|_u, i|_{ls}, E_{k^+|_{ls}}(i_{i1,ls}|_u, \mathbf{n}|_u)); \quad (4)$$

$$ip|_{idp2} \rightarrow ip|_{sp} : \text{MS}_{k^-|_{idp2}}(\mathbf{isess}|_u, d_6|_u) \quad (5)$$

Figure 15: Formalisation of linking service model: state  $\{\mathcal{C}_a^s\}_{a \in \mathcal{A}}$  and (valid) trace Scenario

in the linking service model; the registration and service provision phases correspond to Figures 3(a) and 3(b), respectively. To prove authenticity, the identity providers sign information for *bs* using their private key. *bs* forwards the authentication assertion from *ii* to *ls* and *is* to prove that the user has authenticated. The referrals by *ii* and *is* include random nonces  $\mathbf{n}|_u, \mathbf{n}'|_u$  to ensure that *bs* cannot link different sessions by comparing them.

The linking service model aims to satisfy a privacy requirement specifically about the linking service, which we call *LS attribute undetectability* (LD). We can express this requirement formally in a similar way to the SID, SPD, and ID requirements:  $\mathcal{C}_{ls} \not\models d_1|_u^* \wedge \dots \wedge \mathcal{C}_{ls} \not\models d_6|_u^*$ .

The linking service model in general is independent from message formats. However, the authors also present an instantiation using the SAML 2.0 [25] and Liberty ID-WSF 2.0 [42] standards. Our model captures that instantiation.

### 4.2.3 Identity Mixer

The formalisation of the case study using Identity Mixer (Section 2.4.3) is shown in Figure 16. The top part defines the initial state  $\mathcal{C}^0$ ; the most notable aspect of it is Alice's secret identifier  $i|_{al}$ . The bottom part of Figure 16 models the communication in Identity Mixer: registration as in Figure 4(a) and service provision as in Figure 4(b). For our purposes, we can represent the commitment to Alice's secret identifier in the first message by a hash  $\mathcal{H}(i|_{al}^\pi, n_{c1,1}|_u^\pi)$ . By inference rule ( $\vdash\text{EI}_3$ ), Alice learns a creden-

$$\begin{aligned}
\mathcal{C}_{al}^0 &= \mathcal{C}_{al}^s \cup \{i|_{al}, n_{c1,1}|^\pi, n_{c1,2}|^\pi, n_{c1,3}|^\pi, n_{c1,7}|^\pi, n_{c2,1}|^\eta, n_{c2,2}|^\eta, n_{c2,3}|^\eta, n_{c2,7}|^\eta, \\
&\quad \mathbf{n}_v|^\zeta, cnd|^\zeta, \mathbf{n}|^\zeta, \mathbf{n}_{1,1}|^\zeta, \mathbf{n}_{1,2}|^\zeta, \mathbf{n}_{1,3}|^\zeta, \mathbf{n}_{1,a}|^\zeta, \mathbf{n}_{2,1}|^\zeta, \mathbf{n}_{2,a}|^\zeta, \\
&\quad \mathbf{n}_v|^\xi, cnd|^\xi, \mathbf{n}|^\xi, \mathbf{n}_{1,1}|^\xi, \mathbf{n}_{1,2}|^\xi, \mathbf{n}_{1,3}|^\xi, \mathbf{n}_{1,a}|^\xi, \mathbf{n}_{2,1}|^\xi, \mathbf{n}_{2,a}|^\xi\} \\
\mathcal{C}_{ii}^0 &= \mathcal{C}_{ii}^s \cup \{n_{c1,4}|^\pi, n_{c1,5}|^\pi, n_{c1,6}|^\pi\}; \\
\mathcal{C}_{is}^0 &= \mathcal{C}_{is}^s \cup \{n_{c2,4}|^\eta, n_{c2,5}|^\eta, n_{c2,6}|^\eta\}; \\
\mathcal{C}_{bs}^0 &= \mathcal{C}_{bs}^s \cup \{\mathbf{n}_{1,b}|^\zeta, \mathbf{n}_{2,b}|^\zeta, \mathbf{n}_{1,b}|^\xi, \mathbf{n}_{2,b}|^\xi\}
\end{aligned}$$

$$\text{Scenario} := \text{Reg}_1|^\pi; \text{Reg}_2|^\eta; \text{ServProv}|^\zeta; \text{ServProv}|^\xi$$

$$\text{Reg}_1 :=$$

$$\mathbf{ip}|_u \rightarrow ip|_{idp1} : \mathcal{H}(i|_u, n_{c1,1}|.); \quad (1)$$

$$\mathbf{ip}|_u \mapsto ip|_{idp1} : \text{ICred}_{k^-|_{idp1}}^{i|_u} (i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u; \{n_{c1,i}|.\}_{i=1}^7) \quad (2)$$

$$\text{Reg}_2 :=$$

$$\mathbf{ip}|_u \rightarrow ip|_{idp2} : \mathcal{H}(i|_u, n_{c2,1}|.); \quad (3)$$

$$\mathbf{ip}|_u \mapsto ip|_{idp2} : \text{ICred}_{k^-|_{idp2}}^{i|_u} (d_5|_u, d_6|_u; \{n_{c2,i}|.\}_{i=1}^7) \quad (4)$$

$$\text{ServProv} :=$$

$$\mathbf{ip}|_u \rightarrow ip|_{sp} : \mathcal{H}(i|_u, \mathbf{n}|.), \mathcal{H}(i_{ii}|_u, \mathbf{n}_{1,2}|.), \mathcal{H}(d_2|_u, \mathbf{n}_{1,1}|.), \mathcal{H}(d_3|_u, \mathbf{n}_{1,3}|.), \quad (1)$$

$$d_1|_u, d_2 > 60|_u, cnd|., k^+|_{tp}, E_{k^+|_{tp}}(i_{ii}|_u, \mathbf{n}_v|.)_{cnd|.};$$

$$\mathbf{ip}|_u \mapsto ip|_{sp} : \text{ZK}(\text{cred}_{k^-|_{idp1}}^{i|_u} (i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u; n_{c1,2}|., n_{c1,5}|.), i|_u, i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u, \quad (2)$$

$$\mathbf{n}|., \mathbf{n}_{1,2}|., \mathbf{n}_{1,1}|., \mathbf{n}_{1,3}|.;$$

$$\mathcal{H}(i|_u, \mathbf{n}|.), \mathcal{H}(i_{ii}|_u, \mathbf{n}_{1,2}|.), \mathcal{H}(d_2|_u, \mathbf{n}_{1,1}|.), \mathcal{H}(d_3|_u, \mathbf{n}_{1,3}|.), d_1|_u,$$

$$k^+|_{idp1}, k^+|_{tp}, E_{k^+|_{tp}}(i_{ii}|_u, \mathbf{n}_v|.)_{cnd|.}; d_2 > 60|_u; \mathbf{n}_{1,a}|., \mathbf{n}_{1,b}|.);$$

$$\mathbf{ip}|_u \rightarrow ip|_{sp} : \mathcal{H}(i|_u, \mathbf{n}|.), \mathcal{H}(d_5|_u, \mathbf{n}_{2,1}|.), d_6|_u, cnd|.; \quad (3)$$

$$\mathbf{ip}|_u \mapsto ip|_{sp} : \text{ZK}(\text{cred}_{k^-|_{idp2}}^{i|_u} (d_5|_u, d_6|_u; n_{c2,2}|., n_{c2,5}|.), i|_u, d_5|_u, d_6|_u, \mathbf{n}|., \mathbf{n}_{2,1}|.; \quad (4)$$

$$\mathcal{H}(i|_u, \mathbf{n}|.), \mathcal{H}(d_5|_u, \mathbf{n}_{2,1}|.), d_6|_u, k^+|_{idp2}; \emptyset; \mathbf{n}_{2,a}|., \mathbf{n}_{2,b}|.)$$

Figure 16: Formalisation of Identity Mixer: state  $\{\mathcal{C}_a^s\}_{a \in \mathcal{A}}$  and (valid) trace Scenario

tial from the issuing protocol linking her attributes to her secret identifier. For instance, from message (2) she can derive

$$\text{cred}_{k^-|_{idp1}}^{i|_u} (i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u; n_{c1,2}|., n_{c1,5}|., n_{c1,6}|.)|^\pi.$$

Note that this credential contains Alice's identifier  $i_{ii}|_u|^\pi$  as an additional attribute: it is used later for anonymity revocation.

In the first message of service provision, again we represent the commitments to Alice's secret identifier and attributes by hashes. For anonymity revocation purposes, the first message additionally includes an encryption of the identifier  $i_{ii}|_u|^\pi$  for the trusted third party, with a condition  $cnd|.$  attached describing when the anonymity of the transaction may be revoked. The ZK proof in message (2) convinces  $bs$  that:

Scheme	AX	AR	SID	SPD	ID	IM	ISM	SL	IL	IIL	ISL
Smart certificates	✓	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗
Linking service model	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
Identity Mixer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓ <sup>†</sup>
Smartcard scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7: Comparison of privacy requirements claimed and satisfied by the various systems. Filled check-mark: satisfied and claimed; empty check-mark: satisfied and not claimed; filled cross: not satisfied and claimed; empty cross: not satisfied and not claimed (see Table 2) . <sup>†</sup>: may not be satisfiable efficiently depending on non-privacy-related requirements.

- Alice owns a credential, signed with  $ii$ 's private key;
- the secret identifier and attributes in the credential correspond to the values or commitments sent previously;
- the property  $d_2 > 60|_u$  is satisfied;
- the encrypted message sent previously is encrypted using  $k^+|_{ttp}$  and contains the identifier in the credential.

The second ZK proof is similar. Note that the commitment  $\mathcal{H}(i|_u, \mathbf{n}|.)$  in messages (1) and (3) is the same, guaranteeing  $bs$  that the two certificates are indeed of the same user.

#### 4.2.4 Smartcard Scheme

The Smartcard scheme (Section 2.4.4) is formalised in Figure 17. The top part of the figure defines the initial state  $\mathcal{C}^0$ . In this system, the user's personal information is exchanged on her behalf by a tamper-resistant smartcard. The smartcard is modelled as actor  $al$ . The smartcard has a certified private key; however, this private key is shared between different smartcards so it does not identify the user. Instead, the smartcard has a secret user identifier  $i|_{al}$ , generated on the card, which is used to generate pseudonyms. The actors  $ii$ ,  $is$ , and  $bs$  each have a private key and a corresponding public key certificate signed by the certification authority.

Figure 17 (bottom) shows the information flows in the Smartcard scheme: registration corresponds to Figure 5(a); service provision to Figure 5(b). Parties derive a shared session key using authenticated key agreement based on public key certificates and exchanged randomness. The smartcard generates pseudonyms of Alice with respect to the two identity providers using hashes. In the service provision phase,  $\mathbf{q}|$  and  $\mathbf{dm}|$  represent  $bs$ 's query: what information it needs, and how recent it should be.

Note that [72] does not specify the exact format of the encrypted message to the trusted third party for anonymity revocation. We chose an encryption of the user's identifier at the address provider because this is most appropriate for our scenario. [72] also does not specify how attributes are sent to the smartcard for caching; we chose to add one additional message to the registration phase containing all attributes.

### 4.3 Results

Table 7 presents the results of the analysis of the four reviewed systems with respect to the requirements in Table 6. The results have been obtained using our Prolog imple-



$$\begin{aligned}
\mathcal{C}_{al}^0 &= \mathcal{C}_{al}^s \cup \{k^+|_{ca}, \text{MS}_{k^-|_{ca}}(i_c|_{\cdot}, k_c^+|_{\cdot}, n_c|_{\cdot}), k_c^-|_{\cdot}, i|_{al}, \mathbf{n}_a|_{\cdot}^{\pi}, \mathbf{n}_a|_{\cdot}^{\eta}, \\
&\quad \mathbf{n}_v|_{\cdot}^{\zeta}, \mathbf{isess}|_{\mathbf{u}}, \mathbf{n}_a|_{\cdot}^{\zeta}, \mathbf{n}_v|_{\cdot}^{\xi}, \mathbf{isess}|_{\mathbf{u}}, \mathbf{n}_a|_{\cdot}^{\xi}\}; \\
\mathcal{C}_{ii}^0 &= \mathcal{C}_{ii}^s \cup \{k^+|_{ca}, \text{MS}_{k^-|_{ca}}(i|_{ii}, k^+|_{ii}, n_{ii}|_{\cdot}), \mathbf{n}_b|_{\cdot}^{\pi}\}; \\
\mathcal{C}_{is}^0 &= \mathcal{C}_{is}^s \cup \{k^+|_{ca}, \text{MS}_{k^-|_{ca}}(i|_{is}, k^+|_{is}, n_{is}|_{\cdot}), \mathbf{n}_b|_{\cdot}^{\eta}\}; \\
\mathcal{C}_{bs}^0 &= \mathcal{C}_{bs}^s \cup \{k^+|_{ca}, \text{MS}_{k^-|_{ca}}(i|_{bs}, k^+|_{bs}, n_{bs}|_{\cdot}), \mathbf{n}_b|_{\cdot}^{\zeta}, \mathbf{dm}|_{\cdot}^{\zeta}, \mathbf{q}|_{\cdot}^{\zeta}, \mathbf{n}_b|_{\cdot}^{\xi}, \mathbf{dm}|_{\cdot}^{\xi}, \mathbf{q}|_{\cdot}^{\xi}\}
\end{aligned}$$

$$\text{Scenario} := \text{Reg}_1|_{\cdot}^{\pi}; \text{Reg}_2|_{\cdot}^{\eta}; \text{ServProv}|_{\cdot}^{\zeta}; \text{ServProv}|_{\cdot}^{\xi}$$

$\text{Reg}_1 :=$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{idp1} : \text{MS}_{k^-|_{ca}}(i_c|_{\cdot}, k_c^+|_{\cdot}, n_c|_{\cdot}), \mathbf{n}_a|_{\cdot}; \quad (1)$$

$$ip|_{idp1} \rightarrow \mathbf{ip}|_{\mathbf{u}} : \text{MS}_{k^-|_{ca}}(i|_{idp1}, k^+|_{idp1}, n_{idp1}|_{\cdot}), \mathbf{n}_b|_{\cdot}; \quad (2)$$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{idp1} : E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{idp1}, \mathbf{n}_b|_{\cdot}) (\mathcal{H}(i|_{\mathbf{u}}, i|_{idp1})); \quad (3)$$

$$ip|_{idp1} \rightarrow \mathbf{ip}|_{\mathbf{u}} : E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{idp1}, \mathbf{n}_b|_{\cdot}) (d_1|_{\mathbf{u}}, d_2|_{\mathbf{u}}, d_3|_{\mathbf{u}}) \quad (4)$$

$\text{Reg}_2 :=$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{idp2} : \text{MS}_{k^-|_{ca}}(i_c|_{\cdot}, k_c^+|_{\cdot}, n_c|_{\cdot}), \mathbf{n}_a|_{\cdot}; \quad (5)$$

$$ip|_{idp2} \rightarrow \mathbf{ip}|_{\mathbf{u}} : \text{MS}_{k^-|_{ca}}(i|_{idp2}, k^+|_{idp2}, n_{idp2}|_{\cdot}), \mathbf{n}_b|_{\cdot}; \quad (6)$$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{idp2} : E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{idp2}, \mathbf{n}_b|_{\cdot}) (\mathcal{H}(i|_{\mathbf{u}}, i|_{idp2})); \quad (7)$$

$$ip|_{idp2} \rightarrow \mathbf{ip}|_{\mathbf{u}} : E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{idp2}, \mathbf{n}_b|_{\cdot}) (d_5|_{\mathbf{u}}, d_6|_{\mathbf{u}}) \quad (8)$$

$\text{ServProv} :=$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{sp} : \text{MS}_{k^-|_{ca}}(i_c|_{\cdot}, k_c^+|_{\cdot}, n_c|_{\cdot}), \mathbf{n}_a|_{\cdot}; \quad (1)$$

$$ip|_{sp} \rightarrow \mathbf{ip}|_{\mathbf{u}} : \text{MS}_{k^-|_{ca}}(i|_{sp}, k^+|_{sp}, n_{sp}|_{\cdot}), \mathbf{n}_b|_{\cdot}; \quad (2)$$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{sp} : E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{sp}, \mathbf{n}_b|_{\cdot}) (\mathbf{isess}|_{\mathbf{u}}); \quad (3)$$

$$ip|_{sp} \rightarrow \mathbf{ip}|_{\mathbf{u}} : \mathbf{isess}|_{\mathbf{u}}, E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{sp}, \mathbf{n}_b|_{\cdot}) (\mathbf{dm}|_{\cdot}); \quad (4)$$

$$ip|_{sp} \rightarrow \mathbf{ip}|_{\mathbf{u}} : \mathbf{isess}|_{\mathbf{u}}, E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{sp}, \mathbf{n}_b|_{\cdot}) (\mathbf{q}|_{\cdot}); \quad (5)$$

$$\mathbf{ip}|_{\mathbf{u}} \rightarrow ip|_{sp} : E'_{\text{AKA}}(k_c^-|_{\cdot}, \mathbf{n}_a|_{\cdot}, k^-|_{sp}, \mathbf{n}_b|_{\cdot}) (d_1|_{\mathbf{u}}, d_2|_{\mathbf{u}}, d_6|_{\mathbf{u}}, E_{k^+|_{lp}}(\mathcal{H}(i|_{\mathbf{u}}, i|_{idp1}), \mathbf{n}_v|_{\cdot})) \quad (6)$$

Figure 17: Formalisation of Smartcard scheme: state  $\{\mathcal{C}_a^s\}_{a \in \mathcal{A}}$  and (valid) trace Scenario

mentation (Section 3.4): for each system, we verified the validity of the trace Scenario from the initial state, and checked which of the requirements hold in the resulting state.

#### 4.3.1 Non-privacy requirements

The two non-privacy requirements *attribute exchange (AX)* and *anonymity revocation (AR)* are satisfied in all systems. Indeed, attribute exchange is the basic requirement of an IdM system. In smart certificates and the linking service model, ISL does not hold. In this case, AR holds automatically because the service provider and identity providers can link service accesses to user profiles (even without the help of the trusted third party). In the two systems satisfying ISL (the Identity Mixer and Smartcard systems),

the transmission of an identifier encrypted for the trusted third party is necessary to fulfil this requirement.

#### 4.3.2 Detectability requirements

The detectability requirements with respect to the service provider, *property-attribute undetectability (SPD)* and *irrelevant attribute undetectability (SID)*, verify the possibility to reveal properties of attributes without revealing the exact value; and to reveal some but not all attributes. In smart certificates, the complete certificate is transmitted, so it satisfies neither requirement. To address SID, the identity provider could issue a separate credential for each user attribute. To partially address SPD, the identity provider could issue several credentials proving common properties of attributes, e.g. an “age > 60” credential. These latter credentials could be obtained during the service provision phase, in effect transforming smart certificates into a relationship-focused system. Indeed, this variant is discussed in [59]. Another possibility is to use certificates that allow efficient proofs of knowledge, as in the Identity Mixer system.

In the linking service model, SPD does not hold. Actually, the linking service model focuses primarily on involvement and linkability issues, leaving the details of the actual attribute exchange to underlying standards. However, in these standards (in particular, SAML) it is not possible to exchange properties of an attribute instead of its value. Recently, an extension to SAML to achieve this has been proposed [56]. Moreover, note that this is a problem in the instantiation of the model with SAML: other instantiations in which SPD does hold may be possible.

IdP attribute undetectability (ID) and LS attribute undetectability (LD) also do not hold in the linking service model. This is because the linking service and the subscription provider both receive the signed authentication assertion from the address provider as guarantee that the user has logged in. However, in the SAML standard, the attributes are part of this signed message, so they also need to be forwarded. Technically, this could be easily solved by signing the attributes separately from the authentication information. Again, this problem is due to the instantiation of the model with SAML. Note that although ID is not explicitly claimed by the other IdM systems, they do satisfy it.

#### 4.3.3 Involvement requirements

The involvement requirements state that an identity provider should not know about the user’s involvement with other identity providers (*mutual IdP involvement undetectability*, IM) or service providers (*IdP-SP involvement undetectability*, ISM). In credential-focused systems, this is natural: the identity provider issues a credential to the user without involving others, and it is not involved in service provisions. Indeed, smart certificates, Identity Mixer and the Smartcard scheme all satisfy IM and ISM.

In the linking service model, ISM does not hold because there is direct communication between the identity providers and the service provider. In a variant of the model [26], the identity providers and service provider communicate indirectly via the linking service. However, here the identity providers encrypt the attributes for the service provider (to preserve privacy with respect to the linking service), and so still need to know its identity. To prevent this, some kind of trusted intermediary (like the smartcard in the Smartcard scheme) seems to be necessary.

Moreover, the linking service model does not satisfy IM. The subscription provider learns from the authentication assertion that the user has an account at the address provider (but not the other way round). This problem is also mentioned in [26]: while

“multiple [identity providers] must give [a service provider] the aggregated set of attributes without knowing about one another’s involvement”, the authors concede that “linked [identity providers] may become aware of just one other [identity provider] – the authenticating [identity provider] – during service provision”. IM can be satisfied (within the standards used) if the subscription provider trusts the linking service to verify the address provider’s signature. Another possibility to satisfy the requirement may be to use group signatures [27] for the authentication assertion from the address provider. This solution prevents the subscription provider from learning at which identity provider the user authenticated, but at the cost of reduced accountability.

#### 4.3.4 Linkability requirements

Finally, we discuss the results for the linkability requirements. *Session unlinkability (SL)* is a natural requirement for relationship-focused systems, because the identity provider generates a new signature over the attributes at every service provision. Indeed, it holds for the linking service model. It also holds for the credential-focused Identity Mixer system because rather than showing the credential (which would allow linking), the user just proves the validity of properties using ZK proofs. In the Smartcard scheme, the smartcard is trusted to correctly send attributes from the credentials it knows. In the smart certificates scheme, however, the complete credential is shown so the requirement is not satisfied. *IdP service access unlinkability (IL)*, in contrast, is natural if the identity provider is not involved in service provision, i.e., for the credential-focused smart certificates, Identity Mixer, and Smartcard schemes. It is less natural for relationship-focused systems such as the linking service model. In this case, private information retrieval [28] can be used so that at least the non-authenticating identity provider does not learn which user he is providing attributes of.

To achieve *IdP profile unlinkability (IIL)* global identifiers should be avoided both in credential-focused and relationship-focused systems. Smart certificates, being based on the user’s public key certificate, do not satisfy this requirement. In Identity Mixer, IIL holds because identity providers do not learn the identifiers of the credentials they issue. In the Smartcard scheme, it holds because each identity provider learns a different identifier based on a secret known only by the smartcard. In the linking service model, the authenticating identity provider generates a session identifier and includes it in the authentication assertion sent to the other identity provider. This forwarding of the assertion can be avoided if identity providers trust the linking service to verify the authentication assertion: identity providers can then issue attributes under different session identifiers, and the linking service can assert the link between them. However, that this only partially solves the problem: identity providers are still both involved in service provision, so they may link using timing information. Indeed, just eliminating global identifiers does not fix IIL in our model.

*IdP-SP unlinkability (ISL)* does not hold for the same two systems that also do not satisfy IIL, and for similar reasons. In smart certificates, all parties learn the user’s public key certificate; in the linking service model, the service provider learns the session identifier from the authenticating identity provider. The other systems satisfy it: in Identity Mixer, not even the issuer of the credential can recognise a ZK proof about it; in the Smartcard scheme, the smartcard ensures that the information flow between identity providers and service providers is restricted to just the attributes.

However, as a consequence of ISL holding, extra work is needed to achieve accountability in two respects. First, a message encrypted to a trusted third party is provided to the service provider to achieve anonymity revocation. Second, although

service providers do not learn a credential identifier, they do need assurance that the credential has not been revoked. In the Smartcard scheme, the suggested solution is to let the smartcard perform a regular revocation check. Similarly, in the Identity Mixer system, credentials can be given a short lifetime and be checked for revocation at re-issuing [18]. In both cases, revocation is not immediate.

For Identity Mixer, two proposals for immediate revocation have been done [22]. The first proposal is to include a serial number in the credential. The credential can be issued so that either the identity provider learns this serial number or not. The former case makes ISL not satisfied. In the latter case, ISL holds but the credential cannot be revoked if the user loses her serial number or does not wish to participate. Depending on the situation at hand, this latter behaviour may not be acceptable. The second proposal is to use a ZK proof that the credential is on a public list of valid credentials [18]. This allows revocation without the user's help while not breaking ISL; however, the user needs to keep track of all revoked credentials in the system, and despite recent advances [18] this may still not be efficient enough. Note that the Smartcard scheme does not support immediate revocation at all.

## 5 Discussion

In this section we discuss two aspects of our analysis: first, whether our analysis covers all relevant requirements (Section 5.1); and second, what effort is needed to apply the analysis to other systems (Section 5.2).

### 5.1 Privacy Requirements

This work analyses requirements of IdM systems relating to the knowledge of personal information. We have elicited the requirements in Table 1 by analysing both IdM systems [8, 26, 72] and taxonomies of privacy requirements [10, 41]. We now consider whether these requirements cover all relevant aspects of IdM systems.

The requirements of Table 1 fall into three categories: detectability, involvement, and linkability requirements. (AX can be seen as a detectability requirement, and AR as a linkability requirement.) This paper presents a model in which these three kinds of requirements can be expressed. We first discuss completeness with respect to properties expressible in the model. Next, we consider requirements that cannot be expressed in the model but may nonetheless be interesting, and discuss how the model may be adapted to analyse them.

Table 8 (at the end of this paper) indicates which formal requirements in Table 6 capture which properties in the formal model. The first group of columns indicates the coalition with respect to which a requirement is defined; the next groups list the detectability, involvement, and linkability aspects that it entails.

First consider detectability requirements. With respect to *bs*, all personal information is required to be either detectable by AX, or undetectable by SID and SPD (except for  $d_7$ , which *bs* can always detect by definition of the case study). Similarly, identity providers can detect attributes they endorse by definition of the case study, but no others by ID. (Undetectability of endorsed attributes would be a requirement for the blind certification [8] feature of the Identity Mixer scheme as discussed in Section 2.4.3.) There are no detectability requirements with respect to *ttp*, or about the transaction details  $d_7$ . In fact, these aspects would not produce relevant results because *ttp* never learns any attributes, and *bs* never communicates any transaction details.

Involvement requirements do not cover *ttp* or *al*: the involvement of *ttp* is publicly known, and Alice’s involvement is covered by linkability. For identity providers, there are involvement requirements about all remaining parties, i.e., the other identity provider and the service provider. Usually, service providers assess trustworthiness of user attributes by considering which identity provider endorsed them; hence we do not regard involvement requirements with respect to the service provider as important. (Among the analysed systems, only the Smartcard scheme would satisfy them.)

Linkability requirements capture associations by coalitions of actors. Clearly, at least *ii* and *is* are needed to associate  $\kappa$  and  $\mu$ ; IIL states that without help of others, they cannot. There is no requirement about when *bs* helps them with this; as it turns out, this help never makes a difference. Linkability between user databases and service provisions is defined with respect to the respective identity providers, and with respect to a coalition of all identity and service providers. Considering other coalitions would not reveal interesting differences in the systems we analyse. Similarly, no requirement involves *ii* or *is* in linking the service provisions to each other; in practice, an identity provider would link service provisions to each other by first linking them to its own user profile, which is covered by IL. Finally, AR requires linking the service provisions to  $\kappa$  and not to  $\mu$ ; this is an arbitrary choice made in the definition of the case study.

The above discussion shows that our requirements capture the interesting differences that our model can express; we now discuss some aspects that it cannot express. First consider requirements related to data minimization. Apart from explicitly transferred information, i.e., the user’s attributes, we analyse one particular kind of implicitly transferred information; namely, involvement requirements. However, other kinds may be of interest as well. For instance, the number of transactions performed by a user may be privacy-sensitive, as may be the mere date and time of certain activities (see, e.g., privacy issues in smart metering systems [62]). Knowledge about numbers of transactions can be expressed in our model; date and time may be appended as “tags” to communication. Concerning linkability minimization, we consider only non-probabilistic linking resulting from common identifiers; however, it would also be interesting to consider statistical linking [36] based on overlapping (but non-identifying) information in profiles or the combination of credentials held by a user. Taking these aspects into account would require a probabilistic associability relation. Finally, we stress that we consider privacy strictly in the sense of minimizing knowledge of personal information; in Section 6 we discuss other aspects of privacy.

## 5.2 Applicability of the Analysis Method

Analysing a new system means formally modelling the communication that takes place. This involves expressing the communication in terms of messages modelled in the three-layer model, and modelling an initial state expressing the knowledge required for the communication (personal information, but also information generated during execution, for instance, session identifiers and nonces). This latter task is mainly a matter of industrious bookkeeping.

The main difficulty in modelling a system is in making sure the model accurately captures the cryptographic primitives that are used. We draw upon our experiences in extending the basic formal model of [71] to the present analysis to give the reader a flavour of what this entails. Some operations are easily expressible in terms of standard primitives. For instance, for our purposes, commitments can be modelled as hashes because they satisfy the same inference rules. Other primitives may have different variants requiring different inference rules. For instance, signature schemes may be

with appendix or with message recovery [53]: in the first case, the message is needed for signature verification, in the second case it is not. Thus, attention is needed to ensure that the appropriate model of a primitive is chosen.

When modelling primitives, it is helpful to look at existing formalisations, e.g. using deductive systems [29, 37] or equational theories [3, 12]: they can usually be translated to the three-layer model. For instance, the formalisation of labelled encryption used in this work is based on [21]. Special attention should be paid to the testing rule. Deductive systems do not usually consider testing; equational theories can include rules for signature verification (e.g., [32]) which translate to testing rules in the three-layer model, but may include only those rules that were relevant to the analysis at hand. Thus, to obtain a complete set of testing rules, one needs to take a lower-level look at the operation of the primitive. In addition, note that existing formalisations (e.g., [21]) may not explicitly model randomness in non-deterministic primitives; however, in our model this is needed because we assume messages to be deterministic.

In some cases, no suitable existing formalisation of a cryptographic primitive may be available. In such a case, the general (security) definition of the primitive (e.g., [30] for ZK proofs) generally suffices for obtaining a description for the language  $\mathcal{L}^c$ . However, different implementations of a primitive may give rise to different inference rules. Thus, to obtain inference rules, one needs to consider the particular implementation used in the protocol under analysis. In our experience, this is feasible. Note that because we are only interested in privacy aspects of the primitives, usually some simplifications can be made. See Appendix A for two examples: ZK proofs and anonymous credentials.

Our Prolog implementation of the analysis method makes it easy to modify, add, and remove inference rules. However, it only supports primitives that satisfy some technical assumptions: (1) rules should be either construction, testing or elimination rules; (2) testing rules exist for all sub-messages that appear as preconditions for elimination rules; and (3) there are no infinite cycles consisting of testing and elimination rules. These assumptions are true for the primitives considered in this work, and we expect that they will also be true for many other primitives used in other IdM systems. A generalisation of our implementation beyond these assumptions is left as future work.

The analysis of additional case studies is straightforward in theory, but involves some work in practice. Our analysis method and its implementation are designed to verify properties of particular elements in a particular trace; both need to be modified for other case studies. However, this task can be lightened by exploiting Prolog's programming features. For instance, our case study involves two traces of service provisions, which are almost the same; in our implementation of the model of the systems, both are generated by one Prolog predicate which takes the variable elements as input. This approach can also be used to generate traces with more service provisions, registrations, or actors, and to generate lists of checks that need to be performed for a given privacy requirement. We expect that this does not raise any serious problems, but leave a systematic approach as future work.

## 6 Related Work

We discuss related work on privacy in identity management (Section 6.1) and formal methods (Section 6.2).

## 6.1 Privacy in Identity Management

The principle of protecting privacy by minimizing the knowledge held by actors in IdM systems is well-established in the literature. It has been recognised as a basic “law of identity” for the design of IdM systems [23]. Hansen et al. [41] argue that privacy-enhancing IdM systems should satisfy a high level of data minimization with user-controlled linkage of personal data, and by default unlinkability of different user actions. Pfitzmann and Hansen [61] define privacy-enhancing identity management as preserving unlinkability between user profiles. Finally, in a general survey, Alpár et al. [4] identifying three main privacy issues in identity management: linkability across domains, identity providers knowing user transactions, and violation of proportionality and subsidiarity (i.e., the exchange of minimal information needed to achieve a certain goal). These three issues correspond to our three kinds of privacy requirements: linkability, involvement and detectability, respectively. In contrast to the vision of minimizing actor knowledge, Landau and Moore have recently argued that preventing service providers from collecting transaction data may not be desirable because it prevents the adoption of IdM systems in practice [48]. This falls into a broader discussion on incentives of participants in IdM systems [5, 24] that is out of scope for this work.

In this work we focus on minimizing knowledge of personal information by technical means; other works address other aspects of privacy. Landau et al. [47] argue that privacy protection can be achieved not just technically, but also by legal and policy means. Hansen et al. [41] argue that apart from ensuring data minimization, privacy-enhancing IdM systems should also make the user aware of what information is exchanged about her and who can link it; and allow the user to control these aspects. Bhargav-Spantzel et al. [11] stress the importance of trust between different parties in identity management, and in particular, trust of the user in other parties’ handling of her personal information. Our method can complement this demand for transparency by providing a precise view on how the choice of IdM system impacts privacy. However, interestingly, recent research in behavioural economics suggests that offering transparency to users might actually reduce their privacy by inducing them to release more information [15].

Compared to general discussions, much fewer assessments of privacy in particular IdM systems have been performed. Several studies include privacy in a more general, high-level comparison of IdM systems [1, 43]. These studies consider a broader concept of privacy than we do, also considering transparency and control aspects. However, the analyses have been performed by hand in an informal and high-level (and thus, subjective) way. Concerning knowledge of personal information, [1] considers three different criteria: “usage of pseudonyms/anonymity”; “usage of different pseudonyms” and “user [is] only asked for needed data” (judged on a yes/no scale). [43] considers two: “directed identity”/“pseudonymous/anonymous use” and “minimal disclosure” (judged on a ++ to -- scale). Conversely, we offer a much more granular comparison that is performed using formal methods, and is thus automated, verifiable, and as a consequence, more objective.

Some formal works on privacy in identity management are available. In [61], privacy-enhancing identity management is defined as preserving unlinkability between different user profiles, and the meaning of linkability and its relationship with related concepts is explored in a semi-formal way. Their informal definitions formed the basis of our original work [70] on representing knowledge of personal information. Other formal work on identity management has mainly focused on safety properties with respect to misbehaving attackers, rather than privacy properties with respect to insiders

who follow the protocol specification. In this context, unlinkability [50, 69] and undetectability [21] properties have been considered for Identity Mixer and related anonymous credential schemes. For SAML [25], a standard for the exchange of identity information between identity and service providers used in the linking service model, secrecy properties have been considered [6]. Our work differs from this latter category in two respects: first, we define properties in a general setting, allowing comparisons between different systems; and second, we distinguish between the roles of different insiders rather than considering one outsider, enabling us to express which (coalitions of) actors can associate or detect certain information, and which cannot.

## 6.2 Formal Methods

Formal methods have been around for many years as an important tool to analyse security of communication in IT systems [3, 17, 52, 60]. Most formal methods rely on two basic ideas: the Dolev-Yao attacker model and state exploration techniques. In the Dolev-Yao attacker model, one considers communication messages using idealised cryptographic primitives, and an attacker who controls some or all communication channels between legitimate parties (meaning that he can insert and suppress messages at will, and fabricate messages based on his observations). The reasoning that the attacker performs to fabricate messages can be described by deductive systems (e.g., [29, 37]) or equational theories (e.g., [3, 12]). State space exploration techniques assess the system security by analysing all possible evolutions of a given system in the presence of a Dolev-Yao attacker. The requirements of a system are then verified by checking whether any of the states that can be reached by the system correspond to an attack (e.g., the attacker knows a secret, or has succeeded in impersonating a legitimate user). Several process algebras [3, 14, 54] provide machinery to perform state space exploration. Other approaches have also been proposed, e.g., using induction [60].

Recently, more and more work has focused on using these techniques for privacy properties, in application domains such as electronic toll collection [31], e-voting [32, 33], RFID systems [16], and Direct Anonymous Attestation [66]. These proposals express privacy in terms of “experiments”: slightly different settings for the execution of the same protocol that should be indistinguishable to an attacker. For instance, in electronic toll collection, an attacker should not be able to distinguish a setting in which a first car takes a left road and a second car takes a right road from a situation in which the first car takes the right road and the second car takes the left road. Similarly, in Direct Anonymous Attestation, an attacker should not be able to distinguish a signature produced by one trusted platform module from a signature produced by another one.

Compared to these existing formal methods, our work differs in two crucial ways. First, instead of looking for attack evolutions for each property separately using state space exploration, we consider one single evolution without misbehaving actors, and define our properties in terms of that. Second, to interpret messages as personal information, we reason about them using our three-layer model and own deductive system instead of standard equational theories or deductive systems. We argued that our deductive system is as expressive as standard ones; it would also be interesting to investigate the formal relation between our properties verified with our method, and analogous experiment-based properties verified with state space exploration.

We now discuss existing work modelling the primitives covered by our deductive system. Labelled encryption is a straightforward extension of normal encryption; our model is similar to the one in [21]. The internals of (incorrect) protocols for authenticated key agreement have over the years proven a popular target for analysis using



formal methods [17, 51, 60]; however, we have not found prior works that formally model the external behaviour of (correct) authenticated key exchange protocols in a larger system.

For ZK proofs, both high-level and low-level formalisations exist. In [50], a low-level model of the operation of ZK proofs is given; however, it cannot be used for knowledge derivation; also, questions have been raised about its technical correctness [21]. Two high-level formalisations of ZK proofs have been proposed [7, 21] that, as ours, allow proofs of a restricted set of properties. The equational theory in [7] models the verification of ZK proofs (as our testing rules); the model of [21] only allows correct ZK proofs to take place and does not express their verification. The latter simplification is not suitable for our method, because verification expresses that an actor learns information in new contexts. Note that both model “signature proofs of knowledge” rather than  $\Sigma$ -proofs; however, our methods can also capture that variant.

Three recent proposals [21, 50, 66] are relevant for our formal model of anonymous credentials. [50] only considers operational aspects of anonymous credentials. [21] models credentials and their showing protocol. The model of credentials is similar to ours, and it includes a rule to obtain a credential from a committed message as in our low-level formalisation (Appendix A.2). The showing protocol is formalised in terms of ZK proofs. However, credential issuing is not considered in [21]. Finally, Smyth et al. [66] model joining and signing protocols for ECC-based Direct Anonymous Attestation, which are very similar to issuing and showing protocols for BM-CL-based anonymous credentials [20]. Although our model is based on a different signature scheme [19] and specified at a higher level, their model of signatures generally corresponds to our model of signatures from committed messages in Appendix A.2.

## 7 Conclusion & Future Work

The contributions presented in this work are threefold. First, we have captured the concept of privacy by data minimization in a detailed and comprehensive set of requirements for IdM systems. Second, we have developed a formal analysis method to verify such requirements. Finally, we have formally analysed and discussed the privacy of four representative IdM systems from the literature.

In Table 1, we have presented a list of eleven detailed requirements for IdM systems which address privacy by data minimization. The requirements have been elicited by analysis of both existing taxonomies and existing IdM systems; to the best of our knowledge, we are the first to provide such a comprehensive and detailed taxonomy of requirements. Two requirements capture information that *should* be learned: the attributes that a service provider needs, and the link between user and service access in case of anonymity revocation. The other requirements capture information that *should not* be learned, divided into three categories: detectability (which actors know which attributes), involvement (who may know who else is involved in the identity exchange), and linkability (which coalitions can link information from different contexts). We have shown that the relevant requirements for these categories have been captured in the model. An interesting extension in this direction is to systematically consider other kinds of privacy aspects, such as knowledge of the number or timings of transactions, and uncertain knowledge of associations based on partially overlapping profiles using probability.

We have presented a three-layer model and deductive system to effectively express how actors learn personal information from communication. The model shows what

information an actor knows, and what information he knows to be about the same person. The main feature of this model is that it captures not only the information itself, but also its contents and the context in which it is used. Our earlier works in this direction [70, 71] considered communication using basic cryptographic primitives only; in this work we have also considered reasoning about attribute properties, additional cryptographic primitives, and cryptographic protocols. In particular, our work presents the first model to reason about knowledge arising from ZK proofs and anonymous credential issuing protocols. We needed to extend our previous work to model the systems analysed in this work; similar extensions may be necessary to analyse other systems. We have provided some advice based on our experiences.

We have used this model to verify the elicited requirements for four representative IdM systems: smart certificates, the linking service model, Identity Mixer, and a system based on smartcards. We analysed these systems against the eleven identified requirements (Table 7). It is worth noting that only 17 of these 44 requirements are mentioned as (parts of) requirements in the design of the respective IdM systems. In one instance, we found such a requirement not to hold (a problem which is also mentioned by the authors of the system themselves). In another instance, we clarified the exact setting in which a requirement holds, which may be unrealistic for performance or accountability reasons. The remaining 27 of the 44 entries of the table do not correspond to requirements explicitly stated by the designers of the IdM systems. In this work, we have established whether they hold or not, leading to a more comprehensive analysis and comparison of IdM systems.

As future work, the scope of the analysis can be extended along several directions. First, besides the privacy aspects already mentioned, the method can be extended to analyse assurance and provability aspects. Second, additional IdM systems like U-Prove [58] and the STORK Platform (<https://www.eid-stork.eu/>) as well as other variants of the systems we considered can be analysed and compared. Finally, to evaluate the performance of the method, we plan to apply it to a real case scenario.

We conclude by remarking that the privacy aspects in Table 7 make up only one of the several factors that need to be considered when choosing the “best” IdM system. In fact, the true design challenge for privacy-enhancing IdM systems lies in achieving privacy while at the same time guaranteeing other security aspects such as confidentiality, integrity, availability, accountability and assurance. However, we hope that the precise and detailed privacy assessment provided by our method contributes to making the privacy factor an important one in the design and selection of IdM systems.

**Acknowledgements** This work is partially supported by STW through project ‘Identity Management on Mobile Devices’ (10522).

## References

- [1] Identity Management Systems (IMS): Identification and Comparison Study. Tech. rep., Independent Centre for Privacy Protection Schleswig-Holstein (2003)
- [2] Sony faces legal action over attack on PlayStation network. BBC News (<http://www.bbc.co.uk/news/technology-13192359>) (2011)
- [3] Abadi, M., Fournet, C.: Mobile Values, New Names, and Secure Communication. In: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL ’01), pp. 104–115. ACM (2001)

- [4] Alpár, G., Hoepman, J.H., Siljee, J.: The Identity Crisis: Security, Privacy and Usability Issues in Identity Management. eprint CoRR cs.CR:1101.0427 (January 2011)
- [5] Anderson, R.: Can We Fix the Security Economics of Federated Authentication? In: Proceedings of the 19th International Workshop on Security Protocols (SPW '11), pp. 25–32. Springer (2011)
- [6] Armando, A., Carbone, R., Compagna, L., Cuellar, J., Abad, L.T.: Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. In: Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering (FMSE '08), pp. 1–10. ACM (2008)
- [7] Backes, M., Maffei, M., Unruh, D.: Zero-Knowledge in the Applied Pi-calculus and Automated Verification of the Direct Anonymous Attestation Protocol. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy (SSP '08), pp. 202–215. ACM (2008)
- [8] Bangerter, E., Camenisch, J., Lysyanskaya, A.: A Cryptographic Framework for the Controlled Release of Certified Data. In: Proceedings of the 12th International Workshop on Security Protocols (SPW '04), pp. 20–42. Springer (2004)
- [9] Bella, G., Paulson, L.: Kerberos Version IV: Inductive Analysis of the Secrecy Goals. In: Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS '98), pp. 361–375. Springer (1998)
- [10] Bhargav-Spantzel, A., Camenisch, J., Gross, T., Sommer, D.: User Centricity: A Taxonomy and Open Issues. *J. Comput. Secur.* **15**(5), 493–527 (2007)
- [11] Bhargav-Spantzel, A., Squicciarini, A.C., Young, M., Bertino, E.: Privacy Requirements in Identity Management Solutions. In: Proceedings of the IEEE International Workshop on Human Computer Interaction 2007 (HCI '07), pp. 694–702. Springer (2007)
- [12] Blanchet, B., Abadi, M., Fournet, C.: Automated Verification of Selected Equivalences for Security Protocols. *J. Log. Algebr. Program.* **75**(1), 3–51 (2008)
- [13] Boneh, D.: Twenty Years of Attacks on the RSA Cryptosystem. *Not. Am. Math. Soc.* **46**(2), 1–16 (1999)
- [14] Boreale, M.: Symbolic Trace Analysis of Cryptographic Protocols. In: Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP '01), pp. 667–681. Springer (2001)
- [15] Brandimarte, L., Acquisti, A., Loewenstein, G.: Misplaced Confidences: Privacy and the Control Paradox. In: Ninth Workshop on the Economics of Information Security (WEIS '10) (2010)
- [16] Brusó, M., Chatzikokolakis, K., den Hartog, J.: Formal Verification of Privacy for RFID Systems. In: Proceedings of the 23rd IEEE Computer Security Foundations Symposium (CSF '10), pp. 75–88. IEEE (2010)
- [17] Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. *ACM Trans. Comput. Syst.* **8**, 18–36 (1990)
- [18] Camenisch, J., Kohlweiss, M., Soriente, C.: An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In: Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography (PKC '09), pp. 481–500. Springer (2009)
- [19] Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: Proceedings of the 3rd International Conference on Security in Communication Networks (SCN '02), pp. 268–289. Springer (2003)
- [20] Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Proceedings of the 24th Annual International Cryptology Conference (CRYPTO '04), pp. 56–72. Springer (2004)

- [21] Camenisch, J., Mödersheim, S., Sommer, D.: A Formal Model of Identity Mixer. In: Proceedings of the 15th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'10), pp. 198–214. Springer (2010)
- [22] Camenisch, J., Sommer, D., Zimmermann, R.: A General Certification Framework with Applications to Privacy-Enhancing Certificate Infrastructures. In: Proceedings of the IFIP TC-11 21st International Information Security Conference (SEC '06), pp. 25–37. Springer (2006)
- [23] Cameron, K.: The Laws of Identity. <http://www.identityblog.com/?p=352> (2006)
- [24] Camp, J.: Identity Management's Misaligned Incentives. *IEEE Secur. Priv.* **8**(6), 90–94 (2010)
- [25] Cantor, S., Kemp, K., Philpott, R., Maler, E. (eds.): Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. <http://saml.xml.org/saml-specifications>. OASIS Standard, 15 March 2005
- [26] Chadwick, D., Inman, G.: Attribute Aggregation in Federated Identity Management. *IEEE Comput.* **42**(5), 33–40 (2009)
- [27] Chaum, D., van Heyst, E.: Group Signatures. In: Proceedings of EUROCRYPT '91, Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT '91), pp. 257–265. Springer (1991)
- [28] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private Information Retrieval. In: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95), pp. 41–50. IEEE (1995)
- [29] Clarke, E.M., Jha, S., Marrero, W.R.: Using State Space Exploration and a Natural Deduction Style Message Derivation Engine to Verify Security Protocols. In: Proceedings of the IFIP TC2/WG2.2,2.3 International Conference on Programming Concepts and Methods (PROCOMET '98), pp. 87–106. Chapman & Hall, Ltd. (1998)
- [30] Cramer, R.: Modular Design of Secure yet Practical Cryptographic Protocols. Ph.D. thesis, Universiteit van Amsterdam (1997)
- [31] Dahl, M., Delaune, S., Steel, G.: Formal Analysis of Privacy for Anonymous Location Based Services. In: Proceedings of the Joint Workshop on Theory of Security and Applications (TOSCA'11), pp. 98–112. Springer (2011)
- [32] Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. *Comput. Secur.* **17**(4), 435–487 (2009)
- [33] Dreier, J., Lafourcade, P., Lakhnech, Y.: A Formal Taxonomy of Privacy in Voting Protocols. Tech. rep., Verimag (2011)
- [34] Duhigg, C.: How Companies Learn Your Secrets. *New York Times* (<http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html>) (2011)
- [35] Erdos, M., Cantor, S.: The Shibboleth Architecture. Tech. rep., Internet2 Consortium (2005). Internet2-mace-shibboleth-arch-protocols-200509
- [36] Fellegi, I.P., Sunter, A.B.: A Theory for Record Linkage. *Am. Statist. Soc.* **64**(328), 1183–1210 (1969)
- [37] Fiore, M., Abadi, M.: Computing Symbolic Models for Verifying Cryptographic Protocols. In: Proceedings of the 14th IEEE workshop on Computer Security Foundations (CSFW '01), pp. 160–173. IEEE (2001)
- [38] Fujisaki, E., Okamoto, T.: Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In: Proceedings of the 17th Annual International Cryptology Conference (CRYPTO'97), pp. 16–30. Springer (1997)
- [39] Fyffe, G.: Addressing the insider threat. *Netw. Secur.* **2008**(3), 11–14 (2008)

- [40] Gorman, A.: UCLA Health System warns patients personal information was stolen. Los Angeles Times (<http://articles.latimes.com/2011/nov/05/local/la-me-ucla-medical-data-20111105>) (2011)
- [41] Hansen, M., Berlich, P., Camenisch, J., Clauß, S., Pfitzmann, A., Waidner, M.: Privacy-Enhancing Identity Management. *Inf. Secur. Tech. Rep.* **9**(1), 35–44 (2004)
- [42] Hodges, J., Kemp, K., Aarts, R., Whitehead, G., (eds.), P.M.: Liberty ID-WSF SOAP Binding Specification. <http://projectliberty.org/>. Version 2.0
- [43] Hoepman, J.H., Joosten, R., Siljee, J.: Comparing Identity Management Frameworks in a Business Context. In: *Proceedings of the 4th IFIP WG 9.2, 9.6, 11.6, 11.7/FIDIS International Summer School*, pp. 184–196. Springer (2008)
- [44] Housley, R., Polk, W., Ford, W., Solo, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280
- [45] Jøsang, A., Pope, S.: User-Centric Identity Management. In: *Proceedings of AusCERT Asia Pacific Information Technology Security Conference (AusCERT2005)*. University of Queensland (2005)
- [46] Kellomäki, S.e.: TAS<sup>3</sup> Architecture. Tech. Rep. Deliveral D2.1, work package WP2, version 17, TAS<sup>3</sup> project (2009)
- [47] Landau, S., Gong, H., Wilton, R.: Achieving Privacy in a Federated Identity Management System. In: *Proceedings of the 13th International Conference on Financial Cryptography and Data Security (FC '09)*, pp. 51–70. Springer (2009)
- [48] Landau, S., Moore, T.: Economic Tussles in Federated Identity Management. In: *Tenth Workshop on the Economics of Information Security (WEIS '11)* (2011)
- [49] Law, L., Menezes, A., Qu, M., Solinas, J., Vanstone, S.: An Efficient Protocol for Authenticated Key Agreement. *Designs, Codes and Cryptography* **28**, 119–134 (2003)
- [50] Li, X., Zhang, Y., Deng, Y.: Verifying Anonymous Credential Systems in Applied Pi Calculus. In: *Proceedings of the 8th International Conference on Cryptology and Network Security (CANS '09)*, pp. 209–225. Springer (2009)
- [51] Lowe, G.: Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR. In: *Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS '96)*, pp. 147–166. Springer (1996)
- [52] Meadows, C.: Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends. *IEEE Sel. Areas Commun.* **21**(1), 44–54 (2003)
- [53] Menezes, A.J., Vanstone, S.A., Oorschot, P.C.v.: *Handbook of Applied Cryptography*, 1st edn. CRC Press, Inc. (1996)
- [54] Milner, R.: *Communicating and Mobile Systems: the  $\pi$ -Calculus*. Cambridge University Press (1999)
- [55] Nanda, A.: A Technical Reference for the Information Card Profile V1.0. <http://msdn.microsoft.com/en-us/library/bb298802.aspx> (2007)
- [56] Neven, G., Preiss, F.S.: Attribute Predicate Profile of SAML and XACML. XACML mailing list, Mar 23 2011 (<http://markmail.org/message/2dha2sqmgni7wpc5>)
- [57] OECD: *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. OECD Publishing (2002)
- [58] Paquin, C., Thompson, G.: U-Prove CTP White Paper. Tech. rep., Microsoft (2010)
- [59] Park, J.S., Sandhu, R.: Smart Certificates: Extending X.509 for Secure Attribute Services on the Web. In: *Proceedings of the 22nd National Information Systems Security Conference (NISSC '99)*, pp. 337–348. US Government Printing Office (1999)
- [60] Paulson, L.C.: The Inductive Approach to Verifying Cryptographic Protocols. *Comput. Secur.* **6**(1-2), 85–128 (1998)

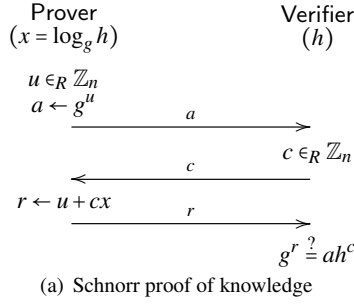
- [61] Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. [http://dud.inf.tu-dresden.de/Anon\\_Terminology.shtml](http://dud.inf.tu-dresden.de/Anon_Terminology.shtml). V0.32
- [62] Rial, A., Danezis, G.: Privacy-Preserving Smart Metering. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society (WPES '11), pp. 49–60. ACM (2011). DOI <http://doi.acm.org/10.1145/2046556.2046564>
- [63] Schnorr, C.P.: Efficient Identification and Signatures for Smart Cards. In: Proceedings of the 9th Annual International Cryptology Conference (CRYPTO '89), pp. 239–252. Springer (1989)
- [64] Seamons, K., Winslett, M., Yu, T., Yu, L., Jarvis, R.: Protecting Privacy during On-Line Trust Negotiation. In: Proceedings of the 2nd International Conference on Privacy Enhancing Technologies (PET '02), pp. 249–253. Springer (2003)
- [65] Smedinghoff, T.J.: Federated Identity Management: Balancing Privacy Rights, Liability Risks, and the Duty to Authenticate. SSRN eLibrary (2009)
- [66] Smyth, B., Ryan, M., Chen, L.: Formal analysis of anonymity in Direct Anonymous Attestation schemes. In: Proceedings of the 8th International Workshop on Formal Aspects of Security & Trust (FAST2011). To appear
- [67] Sommer, D., Mont, M.C., Pearson, S.: PRIME Architecture V3. Tech. rep., PRIME consortium (2008). Version 1.0
- [68] Spiekermann, S., Cranor, L.F.: Engineering Privacy. *IEEE Trans. Software Eng.* **35**(1), 67–82 (2009)
- [69] Suriadi, S.: Strengthening and formally verifying privacy in identity management systems. Ph.D. thesis, Queensland University of Technology (2010)
- [70] Veeningen, M., de Weger, B., Zannone, N.: Modeling identity-related properties and their privacy strength. In: Proceedings of the 7th International Workshop on Formal Aspects of Security & Trust (FAST2010), pp. 126–140. Springer (2011)
- [71] Veeningen, M., Zannone, N., de Weger, B.: Formal Privacy Analysis of Communication Protocols for Identity Management. In: Proceedings of the 7th International Conference on Information Systems Security (ICISS '11), pp. 235–249. Springer (2011)
- [72] Vossaert, J., Lapon, J., De Decker, B., Naessens, V.: User-Centric Identity Management Using Trusted Modules. In: Proceedings of the 7th European Workshop on Public Key Infrastructures, Services and Applications (EuroPKI 2010), pp. 155–170. Springer (2011)
- [73] Waugh, R.: 'Unfair and unwise': Google brings in new privacy policy for two billion users - despite EU concerns it may be illegal. *Daily Mail* (<http://www.dailymail.co.uk/sciencetech/article-2108564/Google-privacy-policy-changes-Global-outcry-policy-ignored.html>) (2012)

## A Inference Rules for Zero-Knowledge Proofs and Credential Issuing

In this appendix we show how our models of ZK proofs and the credential issuing protocol are derived.

### A.1 Zero-Knowledge Proofs

ZK proofs allow a prover to prove to a verifier that he knows some secret information satisfying certain properties with respect to some public information, without revealing any information about the secret. For instance, consider a large group of prime order  $n$



$$p \mapsto v : \text{ZK}(k^-; k^+; \emptyset; \{n_p, n_v\}) \quad (\text{where } k^- = x, k^+ = h, n_p = u, n_v = c)$$

(b) Formal model of Schnorr proof

Figure 18: Schnorr proof of knowledge and its formal model

generated by a group element  $g$ . Note that given value  $h$ , it is infeasible to determine the discrete logarithm  $x = \log_g h$ ; this property can be exploited to build a public key cryptosystem in which values of  $h$  are public keys, and the corresponding values of  $x$  are private keys. A *prover* who knows  $x$  as well as  $n$ ,  $g$ , and  $h$  can engage in a ZK proof protocol with a *verifier* who just knows  $n$ ,  $g$ , and  $h$ ; when the protocol has finished successfully, the verifier is convinced that the prover knows the value of  $x$ , without learning anything about its value.

The general definition of ZK proofs leaves open different kinds of implementations; we model a particular kind of ZK proof called  $\Sigma$ -protocols [30].  $\Sigma$ -protocols are *three-move* protocols in which the prover first sends a *commitment*; the verifier responds with a randomly generated *challenge*; and finally the prover sends a *response*. The ZK proofs used in the systems analysed [8, 19, 20, 38] are of this kind.

An example  $\Sigma$ -protocol is the classical protocol due to Schnorr to prove knowledge of  $x = \log_g h$  in the setting given above (Figure 18(a)). The prover computes a random  $u$  and sends a commitment  $g^u$  to the verifier. The verifier responds with a random challenge  $c$ . The prover calculates response  $r = u + cx$ . The verifier convinces himself that the prover indeed knows the secret  $x$  by checking that  $g^r = ah^c$  using the response, commitment and public information. The prover can only calculate a valid response if he knows the secret; also, the response does not reveal any information about  $x$  [63].

We formally model ZK proofs at a high level using the primitive  $\text{ZK}(m_1; m_2; m_3; n)$ . The secret information  $m_1$  and public information  $m_2$  are described in terms of messages; the ZK proof proves that the public information has a certain message structure with respect to the secret information. In addition, the proof can show that context data items  $d$  occurring in  $m_1$  satisfy properties  $\psi_k(d)$ , listed in  $m_3$ . Finally,  $n$  represents randomness; in  $\Sigma$ -protocols,  $n = \{n_p, n_v\}$ , representing the provers' randomness  $n_p$  for the commitment and the verifier's randomness  $n_v$  for the challenge. For instance,  $\text{ZK}(k^-; k^+; \emptyset; \{n_p, n_v\})$  is a proof of knowledge of the private key  $k^-$  corresponding to public key  $k^+$  with no properties and contributed randomness  $n_p, n_v$ . From this high-level description in terms of structure of messages, the low-level description follows implicitly. For instance, in a setting where public/private key pairs are of the form  $(h, x = \log_g h)$ , the proof  $\text{ZK}(k^-; k^+; \emptyset; \{n_p, n_v\})$  corresponds to a proof of knowledge of the discrete logarithm  $x = \log_g h$  of  $h$  like the Schnorr protocol. Figure 18 shows the Schnorr protocol and its formal model in this setting.

$\frac{\mathcal{C}_a \vdash \text{ZK}(m_1; m_2; m_3; \{n_p, n_v\})}{\mathcal{C}_a \vdash m_3} (\vdash \text{EZ}_1')$	$\frac{\mathcal{C}_a \vdash \text{ZK}(m_1; m_2; m_3; \{n_p, n_v\})}{\mathcal{C}_a \vdash n_v} (\vdash \text{EZ}_2')$
$\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \Rightarrow m_2$	$\frac{\mathcal{C}_a \vdash \text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \quad \mathcal{C}_a \vdash n_p}{\mathcal{C}_a \vdash m_1} (\vdash \text{EZ}_3')$
$\frac{\mathcal{C}_a \vdash \text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \quad \mathcal{C}_a \vdash m_1}{\mathcal{C}_a \vdash n_p} (\vdash \text{EZ}_4')$	$\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \Rightarrow n_p$
$\frac{\mathcal{C}_a \vdash \{m_1, \dots, m_j, n_1, \dots, n_k, p_1, \dots, p_l, n_p, n_v\}}{\mathcal{C}_a \vdash \text{ZK}(\{m_1, \dots, m_j\}; \{n_1, \dots, n_k\}; \{p_1, \dots, p_l\}; \{n_p, n_v\})} (\vdash \text{CZ}')$	

Figure 19: Complete set of inference and testing rules for ZK ( $\mathcal{C}_a$  a set of context messages;  $m_*$ ,  $n_*$  context messages;  $p_i$  properties of  $m_k$ , i.e., every  $p_i = \psi_j(m_k) \in D^c$  for some  $j, k$ )

In Figure 19, we present a set of inference rules for the ZK primitive. We first explain them, and then argue that for privacy purposes and under certain assumptions, it suffices to consider the smaller set of rules presented in Figure 9. We first discuss what messages can be derived from a ZK transcript  $\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\})$  using elimination and testing rules. In general, secrecy of the properties  $m_3$  to be proven is not a privacy goal in the ZK literature; thus, the particular property proven by a ZK proof will influence the format of the messages transmitted. As an over-estimate, we allow any actor to derive the properties  $m_3$  from the transcript ( $\vdash \text{EZ}_1'$ ). The verifier randomness  $n_v$  is transmitted as challenge, and so can be derived from the transcript ( $\vdash \text{EZ}_2'$ ). In particular, any observer who knows the public information  $m_2$  can also verify the ZK proof; hence  $\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\}) \Rightarrow m_2$ . Because both parties are already assumed to know  $m_2$  before the start of a ZK proof, it does not need to follow from the transcript. In fact, it is usually not possible to derive  $m_2$ , as can be seen in the Schnorr example.<sup>3</sup>

The fact that the protocol is zero-knowledge means that a verifier (who knows  $m_2$ ,  $m_3$  and  $n_v$ ) should not be able to learn anything about  $m_1$ . In fact, if there are several possible secrets  $m_1$  corresponding to public information  $m_2$ , then the probability distribution for protocol transcripts is required to be independent from  $m_1$ . Thus, it is impossible to test  $m_1$  from the transcript. (Of course, if  $m_2$  determines  $m_1$ , e.g., if they are a public/private key pair, then  $m_1$  can be derived using  $m_2$ , but this is not due to the ZK proof.) Because the verifier, who knows all components of the ZK proof except  $m_1$  and  $n_p$ , cannot deduce anything about the secret  $m_1$ , any inference rule to derive it needs to have  $n_p$  as a prerequisite. By a similar line of reasoning, if  $m_1$  can be derived from  $n_p$ , then an inference rule for  $n_p$  needs  $m_1$ , or it needs to be a testing rule. In fact, as can be seen from the example of the Schnorr proof, in  $\Sigma$ -protocols all these inferences can be made:  $m_1$  can be derived directly from  $n_p$  ( $\vdash \text{EZ}_3'$ ) and vice versa ( $\vdash \text{EZ}_4'$ ), and  $n_p$  can be tested.

To generate a transcript  $\text{ZK}(m_1; m_2; m_3; \{n_p, n_v\})$  of a  $\Sigma$ -protocol, an actor needs  $n_p$  for the commitment;  $n_v$  for the challenge; and both pieces of randomness and the private information for the response  $n_p$  ( $\vdash \text{CZ}'$ ). (Technically, the public information is not needed.) Similarly, for determinability of the message transmission  $a \mapsto b : \text{ZK}(m_1; m_2; m_3; \{n_p, n_v\})$ , the prover needs  $\{m_1, n_p\}$  in addition to the com-

<sup>3</sup>However, note that if we append  $h$  to the first message of the Schnorr proof, it is still a valid  $\Sigma$  protocol, but now one from which the public information can be derived.



$$\boxed{
\begin{array}{c}
\frac{\mathcal{C}_a \vdash k^+ \quad \mathcal{C}_a \vdash m_1 \quad \mathcal{C}_a \vdash n_a}{\mathcal{C}_a \vdash S_{k^-}^0(m_1, n_a)} (\vdash \text{CS}^0) \quad \frac{\mathcal{C}_a \vdash S_{k^-}^0(m_1, n_a) \quad \mathcal{C}_a \vdash \{k^-, m_2, n_b\}}{\mathcal{C}_a \vdash S_{k^-}(m_1, m_2, n_a, n_b)} (\vdash \text{CS}^{0'})
\end{array}
}$$

Figure 20: Inference rules for signature scheme with signatures on committed values ( $\mathcal{C}_a$  a set of context messages;  $k^+/k^-$  a public/private key pair;  $m_*$ ,  $n_*$  context messages)

munication addresses  $\{a, b\}$ ; the verifier needs  $n_v$ .

There are two aspects the above model does not take into account. First, from two ZK proofs using the same prover randomness, the secret can be derived: in case of the Schnorr proof, by computing  $(r - r')/(c - c')$  from transcripts  $(a, c, r)$  and  $(a, c', r')$ . This is a general property of  $\Sigma$ -protocols called *special soundness*. However, if the prover always honestly generates his randomness, then this is very unlikely and we can safely ignore it. Second, an actor can also “simulate” a ZK proof transcript without knowing the secret information by first generating the challenge and response and from that determining the commitment. Such a simulation has the exact same form as a ZK proof, but because the randomness in the commitment is unknown, it cannot be used to derive a secret corresponding to the public information. Such simulations are very unlikely to correspond to ZK proofs that really took place, so they are not relevant for knowledge analysis.

To express privacy requirements, the knowledge of randomness is not directly relevant. In addition, assuming that the randomness of the ZK proof is freshly generated and not re-used elsewhere, it is clear that it cannot help to derive information indirectly:  $(\vdash \text{EZ}_3')$  is the only rule to derive personal information (namely,  $m_1$ ) using randomness, and it has knowledge of  $n_p$  as prerequisite, which can only be derived when  $m_1$  is already known. Ignoring rules  $(\vdash \text{EZ}_2')$ ,  $(\vdash \text{EZ}_4')$ , we obtain the inference rules given in Figure 9, testability relations in Table 4, and determinability requirements in Table 5.

## A.2 Anonymous Credentials and Issuing

In an anonymous credential system, credentials  $\text{cred}_{k^-}^{M_1}(M_2; M_3)$  assert the link between a user’s identifier  $M_1$  and her attributes  $M_2$  using secret key  $k^-$ , and such credentials are issued and shown anonymously [19]. Anonymous issuing means the issuer of the credential does not learn the user’s identifier  $M_1$  (in particular, this means he cannot issue credentials containing the identifier without the user’s involvement). We model the issuing protocol by the  $\text{ICred}_{k^-}^{M_1}(M_2; M'_3)$  primitive. The randomness  $M'_3$  used in the issuing protocol determines the randomness  $M_3$  in the credential. Anonymous showing means that it is possible to perform ZK proofs of ownership of a credential proving certain properties. This is captured by our ZK primitive.

We model anonymous credential systems constructed from signature schemes [19, 20] as used in the Identity Mixer system [8]. In general, this construction is possible if the signature scheme allows for issuing of signatures on committed values (Figure 20). That is, a commitment  $S_{k^-}^0(m_1, n_a)$  to message  $m_1$  using randomness  $n_a$  is constructed using public key  $k^+$  ( $\vdash \text{CS}^0$ ); this commitment is turned into signature  $S_{k^-}(m_1, m_2, n_a, n_b)$  using private key  $k^-$ , message  $m_2$  and randomness  $n_b$ , ( $\vdash \text{CS}^{0'}$ ). Based on such a scheme, an anonymous credential  $\text{cred}_{k^-}^{m_1}(m_2; \{n_a, n_b\})$  is simply a randomised signature (containing secret identifier  $m_1$  and attributes  $m_2$ ) along with its used randomness. In the Identity Mixer system, two such signature schemes can be

$$\begin{aligned}
a \rightarrow b & : S_{k^-}^0(m_1, n_2); \\
a \mapsto b & : \text{ZK}(m_1, n_1, n_2; k^+, \mathcal{H}(m_1, n_1), S_{k^-}^0(m_1, n_2); \emptyset; \{n_3, n_4\}); \\
b \rightarrow a & : \{S_{k^-}(m_1, m_2, n_2, n_5), n_5\}; \\
b \mapsto a & : \text{ZK}(k^-; k^+, S_{k^-}^0(m_1, n_2), m_2, n_5, S_{k^-}(m_1, m_2, n_2, n_5); \emptyset; n_6, n_7)
\end{aligned}$$

Credential obtained:  $\{S_{k^-}(m_1, m_2, n_2, n_5), n_2, n_5\}$   
(a) Issuing protocol for anonymous credentials

$$a \mapsto b : \text{ICred}_{k^-}^{m_1}(m_2; \{n_i\}_{i=1}^7)$$

Credential obtained:  $\text{cred}_{k^-}^{m_1}(m_2; \{n_2, n_5\})$   
(b) Formal model of anonymous credential issuing protocol

Figure 21: Anonymous credentials from signature scheme with signatures on committed values

used: SRSA-CL signatures [19] and BM-CL signatures [20]. There are slight technical differences between the two; we discuss SRSA-CL signatures and briefly outline the differences later.

The anonymous credential issuing protocol can be modelled as a trace in terms of the signature scheme (Figure 21(a)). It involves a user  $a$  and an issuer  $b$ . As before,  $a$  is assumed to send a commitment  $\mathcal{H}(m_1, n_1)$  to her secret identifier to  $b$  prior to initiating the protocol. (Unlike the commitment  $S_{k^-}^0(m_1, n_2)$  for the signature,  $\mathcal{H}(m_1, n_1)$  does not depend on  $k^-$  and can thus be shared with other issuing or showing protocols for credentials having a different key.) In the first two messages, actor  $a$  provides her commitment for the signature, and then proves that it is formed correctly; that is, it indeed contains the identifier corresponding to the one in  $\mathcal{H}(m_1, n_1)$ . Actor  $b$  uses the commitment to construct a signature on  $\{m_1, m_2, n_2, n_5\}$ , and sends the signature along with his randomness to  $a$ . At this point,  $a$  knows the signature and the two pieces of randomness used in it: these three components together form the anonymous credential, as shown in the Figure. (Note that  $b$  does not know  $n_2$ , so he does not have the complete credential.) In the last step, the signer  $b$  proves that  $S_{k^-}(m_1, m_2, n_2, n_5)$  is valid; when using the SRSA-CL signature scheme, this step is technically needed to ensure the security of the signature [8]. Figure 21(b) displays our high-level model of the issuing protocol and the credential obtained from it.

The high-level inference rules (Figure 10), testability relation (Table 4) and determinability relation (Table 5) for  $\text{cred}$  and  $\text{ICred}$  follow from the lower-level model in Figure 21(a). The credential's signature can be verified using  $\{k^+, m_1, m_2\}$ , and a credential can be constructed from its components ( $\vdash \mathbf{CR}$ ). Although randomness can be inferred from the credential, we do not model these inferences in the high-level model because they are not relevant for knowledge of personal information.

From the issuing protocol, the user can infer the credential using the randomness from the credential ( $\vdash \mathbf{EI}_3$ ). We check the messages of the trace for further possible inferences. For the two  $\text{ZK}$  proofs, ( $\vdash \mathbf{EZ}_1$ ) does not apply because there are no proofs of properties. The ( $\vdash \mathbf{EZ}_2$ ) rule can be applied to both  $\text{ZK}$  proofs occurring in the issuing protocol; this translates to rules ( $\vdash \mathbf{EI}_1$ ) and ( $\vdash \mathbf{EI}_2$ ). We also consider the derivation of the nonces  $n_1, n_2$  ( $\vdash \mathbf{EI}_2$ ):  $n_1$  is generated outside of the issuing protocol, so its derivation may be of interest;  $n_2$  is a prerequisite for ( $\vdash \mathbf{EZ}_3$ ). We do not add a rule to derive  $S_{k^-}^0(m_1, n_2)$  from the transcript because its knowledge is not interesting from a privacy

point of view. However, knowledge of this message does mean that  $\{m_1, k^+, n_2\}$  can be tested. The two testing rules for the ZK primitive each give two additional testing rules for ICred. The third message transmission in the trace allows one to link issued credentials to their issuing protocols.

Finally, consider  $\text{ICred}_{k^-}^{m_1}(m_2; \{n_i\}_{i=1}^7)$ 's determinability requirements. Assuming fresh nonces, determinability of  $\{a, b, k^+, m_1, n_2\}$  by  $a$  is required for the first message transmission. For the first ZK proof,  $a$  additionally requires  $n_1$  and  $n_3$ ;  $b$  requires  $n_4$ . The next message means determinability of  $\{k^-, m_2, n_5\}$  by  $b$ . The last ZK proof additionally means determinability of  $\{k^+, n_6\}$  by  $b$ ;  $a$  requires  $n_7$ . We get the determinability requirements given in Table 5. Note that technically,  $a$  does not need  $m_2$  to run the protocol, and  $b$  does not need  $\mathcal{H}(m_1, n_1)$ ; however, in practice, they will check whether the data supplied matches their expectations using the checks expressed by the testing rules.

We mention two modelling details regarding the use of SRSA-CL signatures for anonymous credentials. First, the last ZK proof in the issuing trace is technically not a proof of knowledge of the private key, but of the RSA inverse of part of the issuer's randomness. However, in terms of knowledge this proof is equivalent because the private key can be determined from the RSA inverse and vice versa [13]. Second, due to the structure of the signature, different choices for  $n_a$  and  $n_b$  can lead to content equivalent signatures. However, assuming  $n_a$  and  $n_b$  are chosen at random, this happens with negligible probability.

Finally, an alternative signature scheme supporting signatures on committed values is the BM-CL scheme [20]. There are two technical differences with the SRSA-CL-based system presented above. First, BM-CL signatures have the additional property that they allow "blinding": a user can turn a valid credential  $\text{cred}_{k^-}^{m_1}(m_2; \{n_a, n_b\})$  into a different credential  $\text{cred}_{k^-}^{m_1}(m_2; \{n'_a, n_b\})$  (however, she is not able to change randomness  $n_b$ ). Second, the final ZK proof in the issuing protocol of Figure 21 is not necessary for a BM-CL-based scheme. We chose the SRSA-CL-based signature scheme because the high-level model is simpler; however, in terms of privacy the choice of signature scheme does not matter.

Requirement	Coalition of...				■: undetectable w.r.t. coalition □: detectable w.r.t. coalition							Involvement unknown				■: unassociable w.r.t. coalition □: associable w.r.t. coalition					
	<i>bs</i>	<i>ii</i>	<i>is</i>	<i>ttp</i>	<i>d<sub>1</sub></i>	<i>d<sub>2</sub></i>	<i>d<sub>2&gt;60</sub></i>	<i>d<sub>3</sub></i>	<i>d<sub>5</sub></i>	<i>d<sub>6</sub></i>	<i>d<sub>7</sub></i>	<i>ii</i>	<i>is</i>	<i>bs</i>	$\kappa, \mu$	$\kappa, \zeta$	$\kappa, \xi$	$\mu, \zeta$	$\mu, \xi$	$\zeta, \xi$	
AX	✓				□		□			□											
SID	✓							■	■												
SPD	✓					■															
ID		✓						■	■	■											
ID			✓		■	■	■														
IM	✓												■								
IM		✓	✓										■								
ISM	✓													■							
ISM		✓												■							
AR	✓	✓	✓	✓												□			□	□	□
SL	✓																			■	
IL		✓														■		■			
IL			✓																■		
IIL		✓	✓																		
ISL	✓	✓	✓	✓																■	

Table 8: Schematic overview of the requirements in Table 6. Each row indicates that with respect to the given coalition of actors, (a) the given items should be (un)detectable; (b) the involvement of the given actors should be unknown; and (c) Alice’s profiles in the given domains should be (un)associable